

Kurt Bauknecht
Sanjay Kumar Madria
Günther Pernul (Eds.)

LNCS 2115

Electronic Commerce and Web Technologies

Second International Conference, EC-Web 2001
Munich, Germany, September 2001
Proceedings



Springer

Lecture Notes in Computer Science

Edited by G. Goos, J. Hartmanis and J. van Leeuwen

2115

Springer

Berlin

Heidelberg

New York

Barcelona

Hong Kong

London

Milan

Paris

Tokyo

Kurt Bauknecht Sanjay Kumar Madria
Günther Pernul (Eds.)

Electronic Commerce and Web Technologies

Second International Conference, EC-Web 2001
Munich, Germany, September 4-6, 2001
Proceedings



Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

Kurt Bauknecht
University of Zürich, IFI
Winterthurer Str. 190, 8057 Zürich, Switzerland
E-mail: baukn@ifi.unizh.ch

Sanjay Kumar Madria
Purdue University, Department of Computer Science
West Lafayette, IN 47907, USA
E-mail: madrias@umr.edu

Günther Pernul
University of Essen, Department of Information Systems
Universitätsstr. 9, 45141 Essen, Germany
E-mail: pernul@wi-inf.uni-essen.de

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Electronic commerce and web technologies : second international conference ;
proceedings / EC Web 2001, Munich, Germany, September 4 - 6, 2001.
Kurt Bauknecht ... (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ;
London ; Milan ; Paris ; Tokyo : Springer, 2001
(Lecture notes in computer science ; Vol. 2115)
ISBN 3-540-42517-9

CR Subject Classification (1998): C.2, H.4, H.3, K.4.4, K.6.5, J.1, J.4

ISSN 0302-9743

ISBN 3-540-42517-9 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

<http://www.springer.de>

© Springer-Verlag Berlin Heidelberg 2001
Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP Berlin, Stefan Sossna
Printed on acid-free paper SPIN 10839841 06/3142 5 4 3 2 1 0

Preface

We welcome you to the Second International Conference on E-commerce and Web Technology (ECWEB 2001) held in conjunction with DEXA 2001 in Munich, Germany. This conference, now in its second year, is a forum to bring together researchers from academia and commercial developers from industry to discuss the state of the art in E-commerce and web technology and explore new ideas.

We thank you all for coming to Munich to participate and debate the new emerging advances in this area. The research presentation and discussion during the conference will help to exchange new ideas among the researchers, developers, and practitioners.

The conference program consists of an invited talk by Hannes Werthner, University of Trento, Italy, as well as the technical sessions. The regular sessions cover topics from XML Transformations and Web Development to User Behavior and Case Studies. The workshop has attracted more than 80 papers and each paper has been reviewed by at least 3 program committee members for its merit. The program committee have selected 31 papers for presentation.

We would like to express our thanks to the people who helped put together the technical program: the program committee members and external reviewers for their timely and rigorous reviews of the papers, the DEXA organizing committee for their help in administrative work and support, and special thanks to Gabriela Wagner for always responding promptly.

Finally, we would like to thank all the authors who submitted papers, those presenting papers, and the attendees who make this workshop an intellectually stimulating event.

We hope you will enjoy this conference and make it a success.

September 2001

Sanjay Kumar Madria
Günther Pernul

Conference Organization

General Chair

Kurt Bauknecht, Switzerland
University of Zurich

Program Chair

Electronic Commerce

Günther Pernul, Germany
University of Essen

Web Technologies

Sanjay Kumar Madria, USA
University of Missouri-Rolla

Program Committee Members

Karl Aberer, EPFL Lausanne, Switzerland
Antonio Badia, University of Louisville, USA
Chaitan Baru, University of California San Diego, USA
Bharat Bhargava, Purdue University, USA
Anjali Bhargava, TRW, USA
Sourav Saha Bhowmick, Nanyang Technological University, Singapore
Martin Bichler, Vienna University of Economics and BA, Austria
Walter Brenner, University of Essen, Germany
Stephane Bressan, National University of Singapore, Singapore
Mike Burmester, Royal Holloway University of London, UK
Wojciech Cellary, The Poznan University of Economics, Poland
Roger Clarke, The Australian National University, Australia
Asuman Dogac, Middle East Technical University, Turkey
Eduardo Fernandez, Florida Atlantic University, USA
Elena Ferrari, University of Milan, Italy
Farshad Fotouhi, Wayne State University, USA
Yongjian, Fu, University of Missouri-Rolla, USA
Rüdiger Grimm, Technical University Ilmenau, Germany
Kamalakar Karlapalem, HKUST, China
Hiroyuki Kitagawa, University of Tsukuba, Japan
Stefan Klein, University of Münster, Germany
Matthias Klusch, DFKI German AI Research Center, Germany
Wolfgang Koenig, University of Frankfurt, Germany
Vijay Kumar, University of Missouri-Kansas City, USA
Karl Kurbel, Europe University Frankfurt (Oder), Germany
Winfried Lamersdorf, University of Hamburg, Germany
George Lausen, University of Freiburg, Germany
Alberto Laender, Federal University of Minas Gerais, Brazil
Ronald M. Lee, Erasmus University, The Netherlands
Tan Kian Lee, National University of Singapore, Singapore
Wang-Chien Lee, Verizon Communications, USA

Qing Li, City University of Hong Kong, China
Ee Peng Lim, Nanyang Technological University, Singapore
Huan Liu, Arizona State University, USA
Heinrich C. Mayr, University of Klagenfurt, Austria
Michael Merz, Ponton GmbH, Germany
Bamshad Mobasher, DePaul University, USA
Mukesh Mohania, Western Michigan University, USA
Gustaf Neumann, Vienna University of Economics and BA, Austria
Wee-Keong Ng, Nanyang Technological University, Singapore
Shojiro Nishio, Osaka University, Japan
Rolf Oppliger, eSECURITY Technologies, Switzerland
Stefano Paraboschi, Politecnico di Milano, Italy
Oscar Pastor, Universidad Politecnica de Valencia, Spain
Evangelia Pitoura, University of Ionia, Greece
Gerald Quirchmayr, University of Vienna, Austria
Kai Rannenberg, Microsoft Research Cambridge, UK
P. Krishna Reddy, University of Tokyo, Japan
Alexander Roehm, University of Essen, Germany
Elke A. Rudensteiner, Worcester Polytechnic Institute, USA
Tomas Sabol, University of Technology Kosice, Slovakia
N. L. Sarda, Indian Institute of Technology, Bombay, India
Peter Scheuermann, Northwestern University, USA
Stephanie Teufel, Université de Fribourg, Switzerland
Paul Timmers, European Commission DG XIII, Belgium
A Min Tjoa, Vienna Technical University, Austria
Aphrodite Tsalgatiidou, University of Athens, Greece
Krishnamurthy Vidyasankar, Memorial University of Newfoundland, Canada
Hans Weigand, Tilburg University, The Netherlands
Richard J. Welke, Georgia State University, USA
Hannes Werthner, University of Trento, Italy
Andrew B. Whinston, University of Texas, USA
Vladimir Zwass, Fairleigh Dickinson University, USA

Additional Reviewers

Fredj Dridi, Yoshiharu Ishikawa, Jouni Markkula, Torsten Priebe, Jahn Rentmeister,
Torsten Schlichting, Bernd Schneider, Anya Sotiropoulou, Dimitrios Theotokis

Table of Contents

Invited Talk

Just Business – Shouldn't We Have Some Fun?	1
<i>H. Werthner; Italy</i>	

Web Software Development

An Object-Oriented Approach to Automate Web Applications Development.....	16
<i>O. Pastor, S. Abrahão, J. Fons; Spain</i>	
Tools for the Design of User Friendly Web Applications	29
<i>N.R. Brisaboa, M.R. Penabad, Á.S. Places, F.J. Rodríguez; Spain</i>	
EProMS: An E-commerce Based Process Model for Cooperative Software Development in Small Organisations	39
<i>A. Rashid, R. Chitchyan, A. Speck, E. Pulvermueller; United Kingdom, Germany</i>	

XML Transformation

Extracting Object-Oriented Database Schemas from XML DTDs Using Inheritance.....	49
<i>T.-S. Chung, S. Park, S.-Y. Han, H.-J. Kim; Korea</i>	
Creating XML Documents from Relational Data Sources	60
<i>C.M. Vittori, C.F. Dorneles, C.A. Heuser; Brazil</i>	
Composition of XML-Transformations.....	71
<i>J. Eder, W. Strametz; Austria</i>	

Electronic Payment

Classification and Characteristics of Electronic Payment Systems.....	81
<i>D. Abrazhevich; The Netherlands</i>	
An E-check Framework for Electronic Payment Systems in the Web Based Environment.....	91
<i>A.R. Dani , P. Radha Krishna; India</i>	

Simulation-, Case Studies

Trader-Supported Information Markets - A Simulation Study.....	101
<i>M. Christoffel, T. Franke, S. Kotkamp; Germany</i>	

An Integrated Framework of Business Models for Guiding Electronic
Commerce Applications and Case Studies 111
C.-C. Yu; Taiwan

Modelling, Design, and Complex Transactions

Models and Protocol Structures for Software Agent Based Complex
E-commerce Transactions 121
G. Wang, A. Das; Singapore

A Multidimensional Approach for Modelling and Supporting Adaptive
Hypermedia Systems 132
M. Cannataro, A. Cuzzocrea, A. Pugliese; Italy

Modelling the ICE Standard with a Formal Language for
Information Commerce 142
A. Wombacher, K. Aberer; Germany, Switzerland

Managing Web Data through Views 154
A.R. Arantes, A.H.F. Laender, P.B. Golgher, A.S. da Silva; Brazil

Security Aspects

Applied Information Security for m-Commerce and Digital Television
Environments 166
S. Katzenbeisser, P. Tomsich; Austria

Flexible Authentication with Multiple Domains of Electronic Commerce 176
K.-A. Chang, B.-R. Lee, T.-Y. Kim; Korea

An Asymmetric Traceability Scheme for Copyright Protection without Trust
Assumptions 186
E. Magkos, P. Kotzanikolaou, V. Chrissikopoulos; Greece

Electronic Negotiation, Trust

An Application Architecture for Supporting Interactive Bilateral Electronic
Negotiations 196
M. Rebstock; Germany

Strategies for Software Agent Based Multiple Issue Negotiations 206
D. Deschner, F. Lang, F. Bodendorf; Germany

Product Catalogs

Automatic Construction of Online Catalog Topologies 216
W.-K. Sung, D. Yang, S.-M. Yiu, W.-S. Ho, D. Cheung, T.-W. Lam; Hong Kong

A Two-Layered Integration Approach for Product Information in B2B E-commerce	226
<i>B. Omelayenko, D. Fensel; The Netherlands</i>	

A Visual One-Page Catalog Interface for Analytical Product Selection	240
<i>J. Lee, P. Wang, H.S. Lee; USA</i>	

Web Site Engineering

Engineering High Performance Database-Driven E-commerce Web Sites through Dynamic Content Caching	250
<i>W.-S. Li, K.S. Candan, W.-P. Hsiung, O. Po, D. Agrawal; USA</i>	

XML Enabled Metamodeling and Tools for Cooperative Information Systems.....	260
<i>C. Nicolle, K. Yétongnon; France</i>	

E-Speak - An XML Document Interchange Engine.....	270
<i>S. Graupner, W. Kim, A. Sahai, D. Lenkov; USA</i>	

User Behaviour

Feature Matrices: A Model for Efficient and Anonymous Web Usage Mining	280
<i>C. Shahabi, F. Banaei-Kashani, J. Faruque, A. Faisal; USA</i>	

Faceted Preference Matching in Recommender Systems.....	295
<i>F.N. Loney; USA</i>	

Pinpoint Web Searching and User Modeling on the Collaborative Kodama Agents.....	305
<i>T. Helmy, S. Amamiya, M. Amamiya; Japan</i>	

Business Models and System Aspects

Analyzing Workflow Audit Trails in Web-Based Environments with Fuzzy Logic.....	315
<i>G. Quirchmayr, B. List, A.M. Tjoa; Australia, Austria</i>	

Using Hypertext Composites in Structured Query and Search.....	326
<i>Z. Qiu, M. Hemmje, E.J. Neuhold; Germany</i>	

Categorizing Distribution Model Scenarios for Online Music	337
<i>W. Buhse; USA</i>	

Author Index	349
---------------------------	-----

Just Business – Shouldn't We Have Some Fun?

Hannes Werthner

eCommerce and Tourism Research Lab
irst-ITC and University of Trento, Italy
E-Commerce Competence Centre (EC3), Vienna, Austria
werthner@itc.it

Abstract. Information Technology has changed our life and society already to such an extent that even our visions cannot exist without technology and its applications. And this change is accelerating. The Web and the related e-commerce phenomena – as Web based e-commerce – is just the “latest” example of this development. But with respect to e-commerce we should distinguish two things: the (today disappointed) expectations with the related “hype”, and that what really happens. However, this raises several serious difficulties: the problem to identify the current stage of development, the definition of the phenomena we speak about, and finally, what does this mean for the (near) future. Thus, this contribution raises more questions than providing answers. In doing so, I will use the travel and tourism industry as an example. And this highlights one important aspect we shouldn't forget: Fun.

1. Introduction

Information Technology has changed and penetrated our life, business and society already so much that even our visions cannot exist without technology and its applications. And this process is accelerating. The Web and the related E-Commerce phenomena – as Web based e-commerce – is just the “latest” example of this development. But with respect to e-commerce we should distinguish two things: the (today disappointed) expectations and the related “hype”, either called e-commerce, e-business, e-economy or m-commerce, and that what really happens.

“Is E-Commerce dead, past its prime, or just resting?” is the title of a recent call for papers for a special issue of the *Journal of IT Theory and Application (JITTA)*. It takes up the problem that the business forecasts have not been fulfilled, exemplified by the difficulties of many dot.com companies, and the related stock market developments. But at the same time online transactions are still (at least in some sectors such as the travel and tourism industry) pointing upwards. But these inconsistencies demonstrate one thing, where e-commerce and the Web are just one, although prominent example: the very close relationship between science and research on one side and business development on the other, the very short development cycles (also in research), the importance of immediate results.

And it highlights two phenomena of our time, constituting the context of e-commerce: acceleration and complexity [1]: we are witnessing rapidly evolving

¹ I thank my colleague Francesco Ricci for his comments on an earlier draft of this article.

technological progress, steadily shrinking time intervals between the introduction of new inventions and innovative products. A further example: today more information is written to digital media than the total cumulative and printing during all of the recorded human history [2]. But acceleration is a historic phenomena: taking the three major technologies of mankind – hunting, farming and industry – each one has grown 100 times faster than its predecessor ([3], based on [4])². This is paralleled by the growth of the so-called knowledge based industries. Knowledge, acquired through investments in research and development as well as education, has become a critical factor and source of competition. As a matter of fact, international R&D spending has grown over the last 15 years and R&D oriented companies have shown a strong performance.

The second phenomenon is complexity, which can also be seen in such a development as globalisation. When trying to identify a single aspect of our society by using a social, economic, ecological, or even a cultural point of view, one realizes an ongoing trend towards organization with a simultaneous growth of interdependencies. There exists a relationship between the growth of organizations and complexity. As in the case of the developments in technology and information processing, these are at the same time cause as well as result of industrial changes. Large organizations are also large information processing systems. The ability to digest information is one of the preconditions for their functioning. In fact, the work of most of them is predominantly in information processing.

The complexity of today's society is correlated with the information processing machinery, which, however, produces also too much information. In that sense IT based information processing increases complexity as well as uncertainty. There is an obvious paradox: on one hand IT increases complexity and of information and on the other hand it appears to be the only means to reduce uncertainty, but implying again more IT applications.

The following chapters discuss shortly the current status of e-commerce, followed by a reflection on different definitions given in the literature and ongoing trends. Chapter 5 looks into the (technological) future, based on the so-called *Ambient Intelligence* vision of European research programmes. The next chapter identifies and discusses some problems in e-commerce related research. Finally, we conclude by identifying challenges for research organisations in this context.

2. Where Are We?

Fig. 1 shows different potential futures for e-commerce [5]. Trajectory A) follows a S-shape growth curve (this was the assumption of many forecasts of the past), whereas trajectory B) assumes rapid acceptance, followed by a sudden fall and then growth again. C) shows slow and steady growth, whereas D) grows rapidly and then declines.

² But this assumes that knowledge advances, but this is not guaranteed, take for example the history of Europe between 1 and 1001. And, with every technology we may disappear faster?

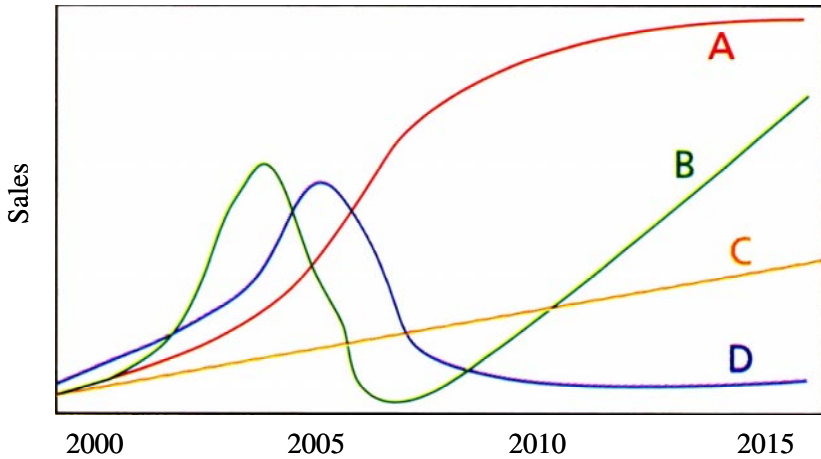


Fig. 1. Qualitative predictions of e-commerce sales

Obviously, stock market performance follows either B) or D).³ But what happens in the “real” economy?⁴ A recent survey⁵ shows cuts in IT expenditure, focusing on fast return of investment. From the interviewed IT professionals nearly half of the respondents said they have cut IT spending, but nearly 66 percent still plan to grow their overall IT budget. Their firms are likely to invest in IT that does show a quick, clear return on investment. And another survey – based on interviews with 150 executives in the US – found that 70 percent said the Internet was essential to the success of their business. Despite the negative reports in the media, the executives had realistic expectations from their e-business initiatives, with Internet sales expected to account for 12 percent of sales, on average, by 2003.⁶

Thus, combining both the stock market performance with the “real” economy, we could have a mixed performance as shown in fig. 2. Obviously, on the long term both trajectories cannot diverge – and still several outcomes are possible.

Numbers published by the US Census Bureau of the Department of Commerce support that – at least in the B2C field – we are in something like a steady state.

The retail e-commerce sales for the first quarter of 2001, not adjusted for seasonal, holiday, and trading-day differences, showed an increase of 33.5 percent with respect to the first quarter of 2000, whereas total retail sales for the first quarter of 2001 increased for 2.3 percent from the same quarter a year ago. However, this online number corresponds to a decrease of 19.3 percent from the fourth quarter of 2000.

³ A report from Tornado-Insider shows a nearly steady level of IPOs for the period Oct. 2000 – March 2001 in the European Union and Israel.

⁴ Not defining the term real economy.

⁵ TechRepublic, In: ACM TechNews Volume 3, Issue 208. May 30, 2001.

⁶ DiamondCluster, In: ACM TechNews Volume 3, Issue 213. June 11, 2001.

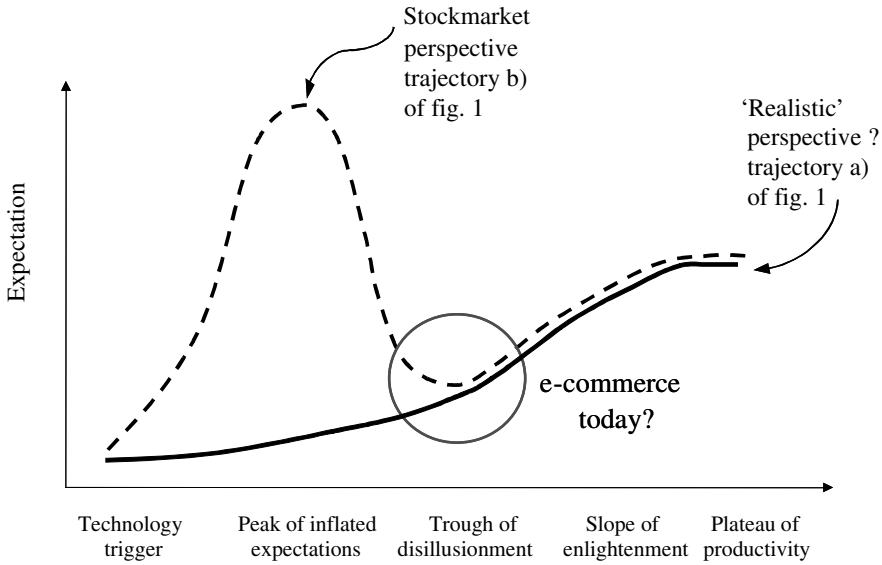


Fig. 2. Stock market versus real world?

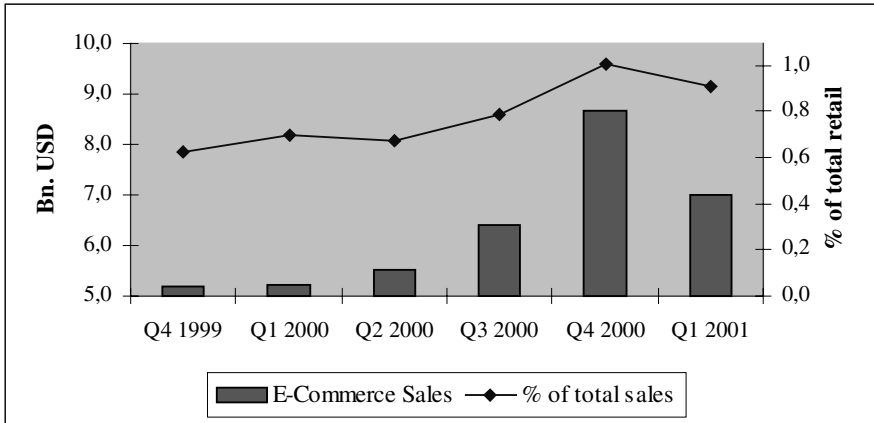


Fig. 3. Retail sales in the US (<http://www.census.gov/mrts/www/current.html>)

Specific sectors perform very well. According to the Travel Industry Association of America over 13 Bn. US\$ were spent online in 2000 for airline tickets, hotel reservations and car rentals in the US (nearly one third of all online B2C spending in the US in 2000). Online bookings in the US increased by 58% in 2000. For Europe a massive jump in the number of Europeans logging on to travel websites is reported, even if in doing so it has left a trail of destruction in the sheer number of B2C

players.⁷ The European Internet Travel Monitor reveals that in 2000 13 Mio. travels in Europe were either initiated or booked via the Internet, which corresponds to 3,5 % of all travels.⁸ Specific sites such as travelocity, expedia or the Austrian destination system TIScover are reporting already profit.⁹

This has to do with the specific features of this sector: it is an information based industry (consumers cannot test the product in advance, but have to rely on the information they receive); it has a world wide consumer as well as supply side; and it represents a networked industry (think of the set of suppliers and intermediaries which have to cooperate to provide packaged product such as a flight together with a hotel and maybe some specific event [6]).

Whereas in other industries there is a stronger hold on to the traditional way things have been done, in the travel industry we are witnessing total acceptance to the extent that the structure of the industry and the way business is conducted, is changing. The net is used not only for information gathering; there is an acceptance of ordering services over the Internet. And it's not the case of only trying one or two services; it is all travel and leisure services. There is a new type of user emerging. The Internet users seem to accept to become their own travel agent – organise their trips themselves and build their own travels and leisure trips.

But there is also another aspect, which has to be taken seriously: travel and tourism has to do with emotional and joyful experiences, with leisure and fun. Even if the Web is not (yet) the highly sophisticated multi media environment, it enables playful experiences, some relaxation as well as the pleasant anticipation of future adventures; it is – overall – a tool for communication and exchange, not (just) business.

These industry features and user behaviour explain why many companies traditionally outside the tourism field are now entering (or have already entered) this sector, take, for example, Microsoft (with Expedia), Bertelsmann or CNN, in cooperation with T-Online and Preussag, the greatest European travel conglomerate.

3. What Are We Talking about?

All these statistics have the problem that they refer to different meanings and are based on varying definitions. Computer scientists normally refer to the technical issues and building blocks – understanding e-commerce as applied computer science, whereas the management science or information system community follows a business and transaction view. There are broad and narrow definitions: either distinguishing between *e-business* and *e-commerce* (seeing the latter as part of the first) or not, in this case both terms are (nearly) interchangeable (see the discussion published in [7]).

On one side you could position e-commerce as “.. is sharing business information, maintaining business relationships, and conducting business transactions by means of

⁷ The eTravel Report – <http://www.travelmole.com>

⁸ This is based on 400.000 interviews in 33 European countries, done in 1999/2000. For example, nearly half of the Swedish population said that they were conducting the Web before making a travel (<http://www.ipkinternational.com>).

⁹ Other examples: BA reports an increase of 40%, whereas Delta Airlines says that online sales already account for 5% of their revenues to business and leisure travellers (FT 12.2.2001).

telecommunication networks” ([8], see also [9] or [10]) with the focus on the coverage of all transaction phases, or the OECD with its definition referring to e-commerce as “business occurring over open, non-proprietary networks (=Internet), including “dedicated” infrastructure; value generating activities within firms, suppliers and customers” [11]. This definition extends to the technology and infrastructure level and equates e-commerce with e-business.

On the other side one can find, for example, [12] with "e-business includes e-commerce but also covers internal processes such as production, inventory management, product development, risk management, finance, knowledge management and human resources. E-business strategy is more complex, more focused on internal processes, and aimed at cost savings and improvements in efficiency, productivity and cost saving" or [13] with the statement that "it is important to note that e-business is much more than electronic commerce. E-business involves changing the way a traditional enterprise operates, the way its physical and electronic business processes are handled, and the way people work". Here e-commerce is viewed as the online exchange of goods, services, and/or money, whereas – on an upper level – e-business automates all business processes and integrates them with e-commerce applications to create one seamless, digital enterprise serving customers and partners.

Thus, three layers within these definitions can be identified, with the obvious interaction of technology and business processes:¹⁰

- Transactions performed,
- Business processes,
- Technology and infrastructure.

But all those definitions fall short in one important aspect as we have seen in the tourism case: they are all very transaction and business oriented and ignore the fact that the Web is a medium of curiosity, of creating communities or of having just fun, all of which may or may not result into business. Where do we have the notion of fun?

4. What Happens?

The evolution of the Web could be described as an ongoing interaction of order and disorder – on different levels:

- Structure with a tendency to concentration and the simultaneous entering of new players (see fig. 4 and 5);
- Services, where, for example, search engines could be identified as tools to create order (at least for the user) and on the other side individualised / recommendation systems (e.g., DoubleClick), or individual pricing (e.g., Priceline); and
- Technology with periods of standardisation, e.g., the work of W3C or IETF, and then (or in parallel) break throughs as now with wireless communication.

¹⁰ It is obvious – given all those different definitions – that the published market research data differ significantly, not to speak about the different methodologies to collect the necessary data since national statistical offices still do not treat this issue. But this is not specific the e-commerce, but common to all new phenomena and transversal sectors.

Fig. 4 (the vertical axis gives portion of web sites on a log-linear scale) demonstrates the tendency to concentration, where a rather small portion of web sites has most of the users. The crucial issue is that sites are rewarded for rather small differences in their relative performances, not on their absolute performance.

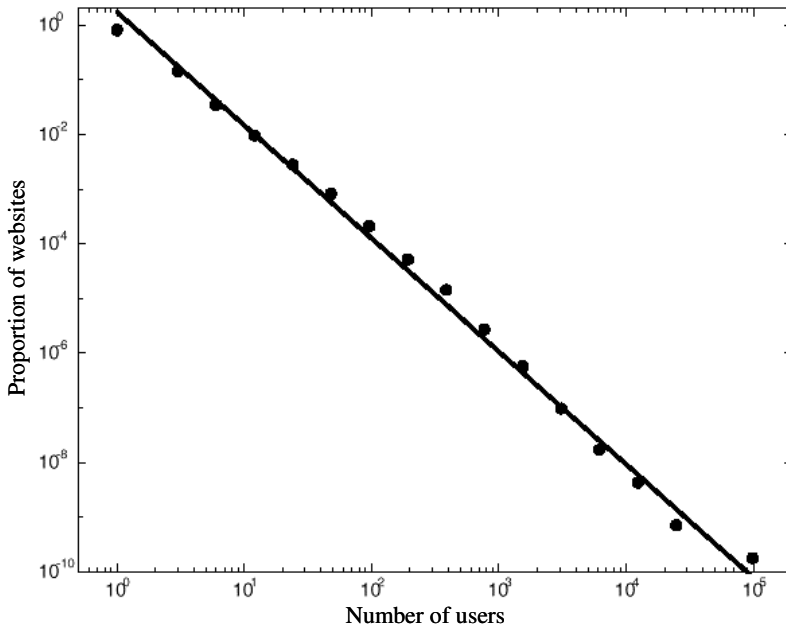


Fig. 4. The winners take it all [14].

At the same time one can observe the permanent appearance of new services. The following example is taken from the tourism industry, where starting from a relatively fixed structure with a already new intermediary in the centre (in this case the Austrian destination information and reservation system TIScover) a new structure emerges featuring intermediate markets: a) one with new intermediaries serving primary suppliers, e.g., hotels, to organise their access to the different new distribution channels; and b) price comparison sites for final consumers comparing different online intermediaries, in this case for a given destination. Obviously, such structures will be “adjusted” given the “rule” shown in fig. 4.

This leads to – or is accompanied by – a “deconstruction” of value chains, where new services tend to become commodities, where with increased quality prices tend to decrease - even for free (e.g., content). With the observable falling of transaction fees this produces rather complex business models¹¹ with income based on other sources (an interesting result, demonstrated in [15], is that those which bundle content or services have a competitive advantage).

¹¹ A business model could be defined as an architecture containing products, services as well as information flows including a description of the various business actors (and a description of the potential benefits for the various business actors) and the respective sources of revenues.

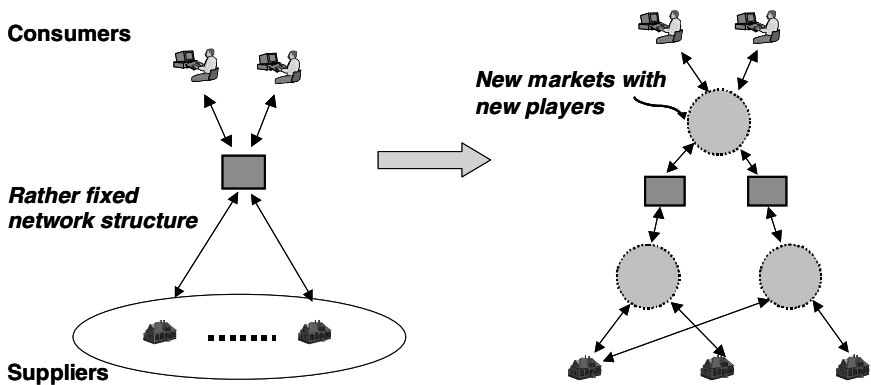


Fig. 5. New structures are emerging

In the travel and tourism industry, as in others such as the media and music industry, the online market was created by newcomers (either start ups or companies from outside the tourism industry), with traditional players such as tour operators or even airlines as careful observers. These stakeholders were constrained by their existing distribution channels (e.g., travel agents) as well as by integration problems into their legacy systems. Now the situation has changed dramatically into one of a hard competitive response, airlines (e.g., Orbitz as a joint venture of US airlines to compete online systems such as Expedia) or tour operators (take the already mentioned example of Preussag) are responding severely. This creates a very competitive situation – both with respect to technological as well as business innovations – including new and rather transient cooperation models between old and new players.

5. A Look into the Future

We have observed the metamorphosis of the computer from a calculator to a media machine, computers changed from being tools to become communication machines, due to transparent technology and access. We have a doubling of computing power every 18 months, of bandwidth every 12 months and IP addresses double every 9 months. And, while we produce 30 Mio. chips per year for PCs, 250 Mio. go to other devices. The PC won't be anymore the major access device, but nearly any human artefact you can imagine, linking them to the Internet. This is exactly the vision of the EU funded research:¹² *Ambient Intelligence*, the surrounding becomes the interface: “We can make huge numbers of inexpensive computing devices which can exchange data very fast; If we could integrate fixed and mobile communication / services in a seamless way; And if we could link these devices to the basis infrastructure and embed them in our surrounding; And if we could incorporate value added services we make the devices to understand the people they serve, we would have an *Ambient Intelligence Landscape*” (www.cordis.lu).

¹² In the so-called framework programmes – starting with FP 6 in 2002.

Ambient Intelligence is the convergence of ubiquitous computing and communication and intelligent user interfaces. Whereas today the dominant mode of interaction is lean-forward (i.e., tense, concentrated), it will become laid-back (i.e., relaxed, enjoyable). People should enjoy, and technology should go to the background. At the end: Why should not your washing machine speak with your dirty linen?¹³

Taking this vision, how could the future look like? This was the exercise performed on behalf of ISTAG (Information Society Technology Advisory Group), describing potential future trajectories or scenarios [16]¹⁴. These scenarios are not good or bad; they serve to provide insights into the technical, social and political aspects of Ambient Intelligence. A series of necessary characteristics permitting the eventual societal acceptance of such technologies were identified such as the facilitation of human contact, the helping to build knowledge and skills for work, citizenship and consumer choice, the need for trust and confidence as well long term sustainability (psychological, societal and environmental). And, such a development should be within human control and enjoyable. The social aspects raise major issues that require precautionary research particularly in the areas of privacy, control and social cohesion.

The main structuring differentials between the scenarios are shown in fig. 6, where these scenarios are not on the same time line (e.g., Carmen seems to be rather near):

- Economic and personal efficiency versus sociability/humanistic drivers;
- Communal versus individual as the user orientation driver.

1. *Maria* is a business lady, travelling via airplane to a business meeting. She uses her personal communicator, her virtual agent has arranged the trip and links permanently to the necessary networks and systems. She can use her finger print to unlock and start the rented car, her P-Com shows the way to the hotel, and the respective room, which also opens automatically, adapts light and temperature to her needs). This scenario is rather incremental and not so distant in time. Crucial issues are privacy, trust as well as security. And there has to be an 'Off-switch'.

2. *Dimitrios* is an employee taking a coffee break, he has his D-Me (maybe embedded in his cloth) and his agent (a learning device) takes phone calls and answers them automatically – giving advice as long as possible and using also Dimitrios' voice. This scenario is on connecting people and experiencing different identities, communication is „bilateral“. We have network-supported relationships; crucial are access privacy, authenticity and ethics.

3. *Carmen* is young woman travelling to her work, disposed to accept vehicle sharing. Waking up in the morning she (her agent) connects to the traffic system, the system tells her the next (private) car passing by, she can – based on an in-car sensor checking whether or not the driver is smoking – decide to accept it. Focus is on traffic optimisation and multi-modal transport. Here we have smart materials and tagging, an advanced traffic infrastructures and urban management services (with all the necessary investments). New behaviour is required as well as – at least to some extent – the acceptance of control (e.g. optimised inter-modality).

¹³ This “European” view is very close to the statements of many authors invited to the special issue of the CACM on “The Next 1000 Years” (CACM 44/3 2001).

¹⁴ ISTAG is the high level advisory group of experts to the European Commission for the IT related research framework programmes.

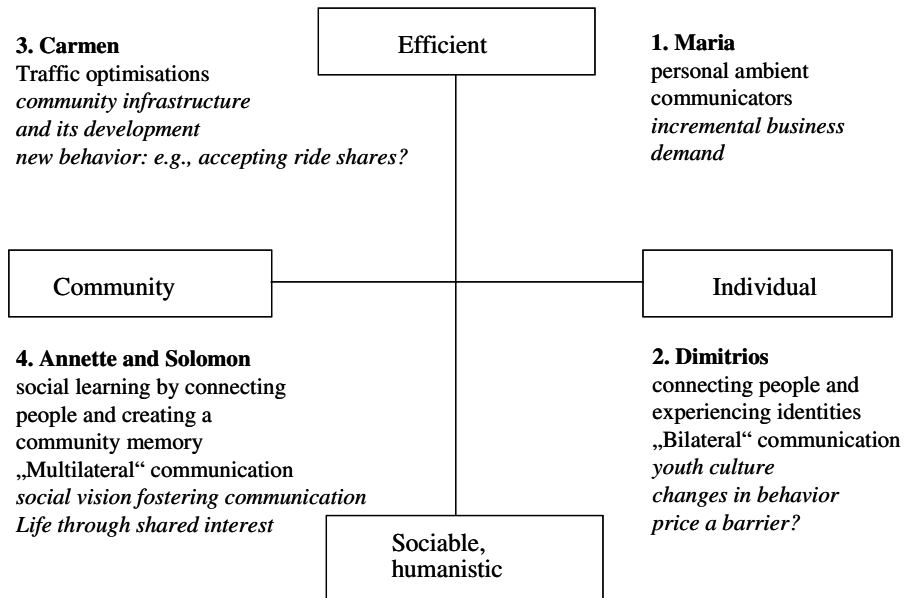


Fig. 6. The four scenarios and their focus

4. *Annette and Solomon*, the last scenario, is about social learning by connecting people and creating a community memory. Communication is „multilateral“, it is based on a vision of life through shared interest. The script uses as an example a plenary meeting of an environmental group, with a real and virtual mentor (agent). Two persons, Annette (known to the group) and Solomon (a new member) are automatically introduced to the group, there are automatic negotiations going on. All different means of communication infrastructure and tools are used. It puts emphasis on the social side, both efficiency and fun are considered. And there is a wide choice as well as personalisation of learning approaches.

It is not assumed that these scenarios will become reality – at least not entirely. They serve to identify potential elements of the future and their very general (technical) requirements, besides the necessary social and economic preconditions described previously, such as¹⁵

- seamless mobile/fixed communications infrastructure,
- very unobtrusive hardware,
- dynamic and massively distributed device networks,
- natural feeling human interfaces,
- privacy, dependability, fault tolerance and security.

¹⁵ Obviously, these are the research areas of EU funded research within the next FP.

6. Issues for Research

The previous look into a (bright) future presumes a well working and functioning technology. However, a serious review uncovers that our (e-commerce) systems are not very “intelligent”, in principle they are more or less advanced well-designed interfaces to databases. Systems are rather complicated – both their architectures and modules as well as the tools used to create them –, and one should not be too confident when using them, given the statistics about software bugs. The list of open issues is very long, on a technical as well as on a methodological level. In the following I will stress just some of them:

6.1. Markets and Users

The tourism domain is an excellent example for the trend towards increasingly personalised services and complex market mechanism providing these services [17]. It reflects that users become part of the product creation process, the trend from *customer focused* to *customer driven*. Customers start to ask for their personal prices. As an example take Priceline using a reversed auction mechanism, where users can define the price they are willing to pay for a product they can partly define, and applying yield management for capacity management.

These market mechanisms are based on auctions and automated negotiations [18], where, in addition to traditional methods, the IT infrastructure enables more complex forms such as multi-attribute or combinatorial auctions. Here the number of (different) goods may vary as well as more than just one attribute (the price) may be used to determine the winner. The design and the analysis of these mechanisms, taking into consideration all the different parameters such as auction type, number of sellers and buyers, matching and scoring rules, number of types of goods or number of attributes used, are rather complex problems [19]. For example, some optimisation problems related to combinatorial auctions are NP complete ([20], [21]). In addition, these specific match making mechanisms are accompanied by an increasing variation of market forms such e-malls, third market places or collaboration platforms [22].

Many of the market related issues are at the interface of different disciplines such as (computational and experimental) economics, econometrics, game theory, computer science as well as simulation. How do these electronic markets work, and how can they be designed and tested. The discussion is about model checking [23] versus market simulation [24]. At the same time economics could use these electronic market places as test beds to check their theories about “real” markets.

Another approach providing individual offers (and prices) is followed by recommendation and collaborative filtering systems. Typically they use case based reasoning to create an offer, based on personal preferences, other user choices and past experiences. [25] describe a project where dynamic product bundling is supported as well. The focus is to facilitate the user interaction, where complex negotiation mechanism may be too complex to be understood – usability is a major issue (see [26] for an interesting approach in experimenting with new market formats and related interfaces).

6.2. Harmonisation

Two different levels can be distinguished: system integration and interoperability, and the semantics issue, also related to the problem of overabundance of information.

Internal and external business processes must be integrated in order for business to be conducted seamlessly. This type of integration means, for example, that a web-based sales order entry system must be integrated with the enterprise's accounting systems so that credit history. The other aspect is the heterogeneous content structure of the Web with no central power instance. At the end both cases can be related to the issue of semantic heterogeneity (assuming that syntactical issues are solved) and meta data models.

Taking again the tourism case as an example, semantic heterogeneity appears on many different levels of service definitions, in addition to the problem of varying terminology, e.g.: a) an overnight might include a breakfast or not, b) room price might be given per person and night or per room and night, c) a double room may be available for one, two or three persons, d) attribute values may vary such as in the case of hotel categories using either stars, crowns, or just integers (and sometimes you may even need the price to identify the category). Mediated or multi-layer architectures, as described by [27], seem to be a promising approach, where specific programme modules –mediators – are dedicated to resolve this issue. But given all these different layers and modules, what will happen with the performance, won't it drop off?

The assumption underlying such harmonisation / mapping of different schema is that in the Web there is no central power instance which is strong enough to impose one (semantic) standard. But such mediators will be very domain dependent, as it can already be seen by the different frameworks proposed (see, for example, [28] for an overview). This raises again an open issue: how to link the different domains?

Typically, XML is used as the basis to describe the respective exchange formats, and RDF as a means to formalize the ontologies or semantic networks, which are at the basis of the domain specific frameworks [29] (see also [30] or [31] for applications in the tourism domain). But, how will we create such ontologies, by more or less formal bodies such as used in the European project Harmonise for the tourism domain,¹⁶ or by a posteriori classification using statistical methods?

6.3. Fault Tolerance

When following the ISTAG vision we will rely on the underlying IT infrastructure, more then ever before. We will need systems that work. Also in the case of a serious fault there must be a guarantee that they function at some assured level. This is most probably the hardest challenge, not only related to e-commerce.

But when looking to the real performance one sees that, for example, systems like eBay, the online auction system, had to be shutdown due to large-scale problems related to defective code. The impact of defective software code on U.S. businesses is reported to be immense, causing firms to lose nearly \$100 billion last year in repair

¹⁶ With the participation of OTA (Open Travel Alliance), TTI (Travel Technology Initiative), ETC (European Travel Commission) and WTO (World Tourism Organisation).

costs, down-time, and lost productivity. A software programmer, on average, makes one mistake per 10 lines of code.⁷

The challenge is in software engineering, testing and code checking, with an obvious contradiction: on one side we have sophisticated software engineering approaches based on formal methods and iterative development cycles, and on the other side the open software approach, with very distributed development procedures and not very formalised rules, produces highly reliable code such as LINUX or Apache.

But how will this work on the long run with its changing and voluntary developer community, especially when time to market becomes shorter and shorter asking for fixed deadlines.

7. As Conclusion – Challenges for Research Organisations

This set of open issue could be easily extended, and it underlines the importance of an interdisciplinary approach, as stated by the National Science Foundation in 1999. Many different disciplines are asked for their contribution, just to mention Computer Science, Management Science, Economics, Law, Statistics, Sociology or Psychology. Many issues are at the interface of these disciplines, who can design a market, simulate and then implement it? Or just think of privacy, which is besides a legal problem also a technical one – at the long run the technology used should prevent misuse. Or what is the impact of specific regulations and could these be simulated, what is the social impact in general. What might happen if a larger social group decides to disconnect, not to participate deliberately – which information they should find when they come back?

The interdisciplinary approach is a major challenge for research, which is based on a very high level of specialisation. Others are, obviously besides excellence, the ability to cooperate, sufficient financial resources – given very insecure public budgets and companies, which want to buy concrete results – and marketing. Research organisation have already realised the need to sell their work, and this will further increase.

How will we treat in such a context high risk and longer term projects, when the cycle of innovation becomes shorter and shorter? Given the varying windows of opportunity in the different areas of research, will we be prepared to push project results to the market, even if there would be still some funding or if the results are not totally perfect? How will we realise the link between innovation and research, which is more critical in Europe than in the USA. How is the support to colleagues who want to start their own business, or the cooperation with industry (pure pull or push models work rarely)? Is there the possibility to experiment with new business models besides going directly to the market? And finally, how should research carriers be designed in an environment, which is based on publications and not on the formulation of business plans or contributions to list-servers.

As you can see, e-commerce raises more questions than providing answers. But maybe this is not specific to e-commerce but in common with all the recent technical and economic developments. Will we be able to create and to manage sustainable and

¹⁷ ACM TechNews Volume 3, Issue 206. May 23, 2001.

robust systems where probably the intelligence is more in the collaboration of different modules than in the modules themselves? The case of tourism teaches us that fun, conversation and enjoyment are essential – not just commerce and business. Maybe, if our future systems won't work, their debugging and reconstruction will make fun.

References

1. Werthner, H., Klein, S.: Information Technology and Tourism. A Challenging Relationship. Springer Verlag, Wien (1999)
2. Fayyad, U.: The Digital Physics of Data Mining. CACM 44/3 (2001) 62-65
3. Varian, H.: The Computer Mediated Economy. CACM 44/3 (2001) 92-93
4. Hansen, R.: Long-Term Growth as a Sequences of Exponential Modes. George Mason University, (1998)
5. Weaver, A., Vetter, R., Whinston, A., Swigger, K.: The Future of E-Commerce. IEEE Computer 33/10 (2000) 30-31
6. Werthner, H., Klein S.: ICT and the changing landscape of global tourism distribution. International Journal of Electronic Commerce & Business Media 9/4 (2000)
7. Alter, S., Ein-Dor, P., Markus, L. Scott, J., Vessey, I.: Does the trend towards e-business call for changes in the fundamental concepts of Information Systems? A Debate. Communications of AIS 5 (2001)
8. Zwass, V.: Electronic Commerce: Structures and Issues. International Journal of Electronic Commerce 1/1 (1996) 3-23
9. Wigand, R. T.: Electronic Commerce: Definition, Theory, and Context. The Information Society 13/1 (1997) 1-16
10. Kalakota, R., Whinston, A. B.: Frontiers of Electronic Commerce. Addison Wesley, Reading (1996)
11. OECD 2001: Business-to-Consumer E-Commerce Statistics. OECD, Berlin (2001)
12. Bartels, A.: The difference between e-business and e-commerce. Computerworld Oct. 30 (2000)
13. El Sawy, O.: Redesigning Enterprise Processes for e-Business. Irwin McGraw-Hill, Boston (2001)
14. Acamic, L., Huberman B.: The Nature of Markets in the World Wide Web. Working Paper. eCommerce Research Forum (<http://ecommerce.mit.edu/forum/>) (1999)
15. Bakos, Y., Brynjolfsson, E.: Aggregation and Disaggregation of Information Goods: Implications for Bundling, Site Licensing and Micropayment Systems. In: Werthner, H., Bichler, M. (eds.): Lecture Series in E-Commerce. Springer Verlag, Wien (2001)
16. Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman, J.-C.: Scenarios for Ambient Intelligence in 2010. Final Report, Version 2. IPTS-Seville (2001)
17. Bichler, M., & Klimesch, R.: Simulation of multi-attribute Auctions (in German). Wirtschaftsinformatik 42/3 (2000)
18. Sandhon, T.: Auctions and automated negotiations. CAMC 24/3 (1999) 84-88
19. Bichler, M. & Werthner, H.: A Classification Framework of Multidimensional, Multi-unit Procurement Negotiations. In: Proc. of the DEXA Workshop on Negotiations in Electronic Markets - beyond Price Discovery - e-Negotiations Workshop. (2000)
20. Rothkopf, M. H., Pekec, A., Harstad, R. M.: Computationally manageable combinatorial auctions. Management Science 44/8 (1998) 1131-1147
21. Hoesel, S. van, Müller, R.: Optimization in Electronic Markets: Examples in Combinatorial Auctions. Netnomics, forthcoming (2001)
22. Timmers, P.: Electronic Commerce. John Wiley, NY (2000)

23. Wang, W., Hidvegi, Z., Baily, A., Whinston, A.: E-Process Design and Assurance Using model checking. *IEEE Computer* 33/10 (2000) 48-54
24. Chaturvedi, A., Mehta, S.: Simulations in Economics and Management. *CACM* 42/3 (1999) 60-62
25. Ricci, F. & Werthner, H.: Case based destination recommendation over an XML data repository. In: Sheldon, P., Fesenmaier D., Wöber, K. (eds.): *Proc. of ENTER Conference 2001*. Springer Verlag, Wien (2001)
26. Bichler, M.: An Experimental Analysis of Multi-Attribute Auctions. *Decision Support Systems* 29/3 (2000)
27. Wiederhold, G., Genereseth, M.: The Conceptual Basis for Mediation Services. *IEEE Expert Intelligent Systems* 12/5 (1997) 38 - 47
28. Shim, S: B2B E-Commerce Frameworks. *IEEE Computer* 33/10 (2000) 40-48
29. Staab, S., Studer, R., Schnurr, H., Sure, Y.: Knowledge Processes and Ontologies. *IEEE Intelligent Systems* 16/1 (2001) 26-35
30. Kaukal, M., Werthner, H.: Integration of Heterogeneous Information Sources. In: Buhalis, D., Fesenmaier, D. (eds.): *Information and Communication Technologies in Tourism. Proc. of the 7th ENTER Conference 2000*. Springer Verlag, Wien (2000)
31. Kaukal, M., Höpken, W., Werthner, H.: An Approach to Enable Interoperability in Electronic Tourism Markets. In: Hansen (ed): *Proc. of the ECIS Conference 2000*

An Object-Oriented Approach to Automate Web Applications Development

Oscar Pastor, Silvia Abrahão and Joan Fons
Department of Information Systems and Computation
Valencia University of Technology

Camino de Vera s/n, P.O. Box: 22012, E-46020 - Valencia, Spain.

e-mail: opastor, sabrahao, jjfons @dsic.upv.es

Abstract. This paper presents the Object-Oriented Web-Solutions Modeling approach (OOWS), which provides mechanisms to deal with the development of hypermedia information systems and e-commerce applications in web environments. It is proposed as an extension of an object-oriented method for automatic code generation based on conceptual models (OO-Method). The main contribution of this work is the introduction of a navigational model that is completely embedded in the process of conceptual modeling, to specify navigational features as a main part of what is conventionally specified during the conceptual modeling process. This navigational model provides abstraction primitives that allow to capture and represent navigational semantics in a precise way. We show how to put into practice the OOWS approach through a successful practical example developed within the context of e-commerce applications.

1 Introduction

Nowadays, with the rapid expansion of the Internet, there are a number of initiatives which are intended to provide a solution for the creation of web applications within a well-defined software production process. Furthermore, these solutions must provide support for e-commerce due to the growth of commercial activities on the network. Several approaches to hypermedia design such as OOHDM [3] EORM [2], HDM [6], OOH-Method [9], W3I3 Tool Suite [1] and ADM [5] have been presented. Normally, hypermedia features and functional properties are dealt with separately, making it difficult to deal with the problem of developing a web application in a unified framework. In practice, they provide only a partial solution because they focus either on hypermedial characteristics (focusing on how the navigation will be defined) or on more conventional characteristics (focusing on classes and operations to specify the system functionality).

Beyond these partial solutions, it is beginning to be widely accepted that web sites are evolving from merely hypermedia information repositories to hypermedia distributed applications, the generally called web applications [10]. Our proposal provides a concrete contribution in this context: starting from claiming that conceptual modeling is needed for developing correct web applications, we introduce a different approach that focuses on the integration of navigational design and conceptual modeling, which together could be used as input for code generation environments.

A software production process to put these ideas into practice has been designed and implemented. This can be called an e-modeling software production environment, meaning that we define a process for applying conceptual modeling techniques to the development of web applications. To properly deal with this problem, we integrate two activities which are traditionally performed separately: that of modeling the operations and that modeling the hypermedia. In our approach, the conceptual modeling step is considered as a unique generic phase. Using what we could call a conventional OO conceptual modeling approach, the needed expressiveness is introduced in the model in order to properly specify navigation and presentation features. All this information is used to provide a precise methodological guidance for going from the conceptual space to the solution space (represented by the final software product).

The method taken as the basis for this approach is the OO-Method [11] [14]. The OO-Method collects the system properties considered relevant for building a formal, textual OO specification in an automated way. This formal OO specification constitutes a high-level system repository. Furthermore, the definition of a concise execution model and the mapping between the specification language and the execution model notions, makes it possible to build an operational implementation of a software production environment allowing for real automated prototyping, by generating a complete system prototype (including static and dynamic characteristics) in the target software development environment. In the context of the OO-Method project, efforts have been oriented towards the development of a new model to enrich the Object-Oriented Software Production Method with the required expressiveness to specify navigation features. This model has been called Navigational Model and provides navigational support for Object-Oriented Web-Solutions Modeling (OOWS) [12].

This paper is organized in four sections. Section 2 describes how the navigational features are represented in conceptual modeling. Section 3 describes the OOWS primitives that incorporate navigational semantics to OO-Method. Section 4 describes a real case study that shows the application of the proposed approach. Finally, section 5 provides concluding remarks and work in progress.

2 Representing Navigational Features in Conceptual Modeling

Following the OO-Method approach, a full specification of the user requirements is built in the conceptual modeling phase. In order to model the desired system including the navigational aspects, we present a web-solution developing process with two major steps: Specifying the System and Developing the Solution.

In the *Specifying the System* step, problem peculiarities and the behaviour that the system must offer to satisfy the user requirements are identified. This step includes the requirements collection with a Use Case [8] approach and system conceptual modeling activities. When dealing with the conceptual modeling phase, the abstractions derived from the problem are specified in terms of classes, their structure, behaviour and functionality. Historically, the OO-Method uses diagrams to represent the required information in three models: the Object Model, the Dynamic Model and the Functional Model.

In this paper, a fourth model is introduced: the so-called Navigational Model. All together, these models describe the object society from four complementary points of view within a well-defined OO framework. The Object Model defines the structure and the static relationships of the classes identified in the application domain. The Dynamic Model defines the possible sequences of services and the aspects related to the interaction between objects. The Functional Model captures the semantic associated to the changes of state of the objects motivated by the service occurrences. In the Navigational Model, the navigation semantics associated to the system users is specified, starting from the classes of the object model. This added expressiveness is the core of what we call the OOWS approach.

In the *Developing the Solution* step, a strategy for components generation to integrate the solution (the final software product) is defined. In this step, a web-based application, which is functionally equivalent to the specification, can be obtained in an automated way.

3 The OOWS Approach

In OOWS, the web-solution application is obtained by adding a navigational view over the OO-Method Object Model. The navigational semantics of a hypermedia application is captured based on the point of view of each agent identified in the object model. This semantics is described in a Navigational Model using a UML-like [4] notation. The navigational model is essentially composed of a **Navigational Map** that represents the global view of the system for an agent in OO-Method. It is represented by a directed graph in which the nodes are the navigational contexts and the arcs are the navigational links.

A **Navigational Context** represents the point of view that a user has on a subset of the Object Model. Basically, it is composed of three areas: context definition, navigational and advanced features. The adopted notation of a navigational context is shown in Figure 1. In this section we explain these areas in details.

A **Navigational Link** is a relationship which is defined implicitly between contexts. This link allows navigating from one context to another.

3.1 Context Definition Area

The context definition area shows the type of context. Navigational contexts can be classified into two types: *exploration contexts* (E) and *sequence contexts* (S). The exploration contexts can be reached at any moment independently of the current context. The sequence contexts can only be reached following a predefined sequence of navigational links.

3.2 Navigational Area

The navigational area is composed by a set of navigational classes which are stereotyped with the reserved word «view». In fact, a *navigational class* can be defined as a

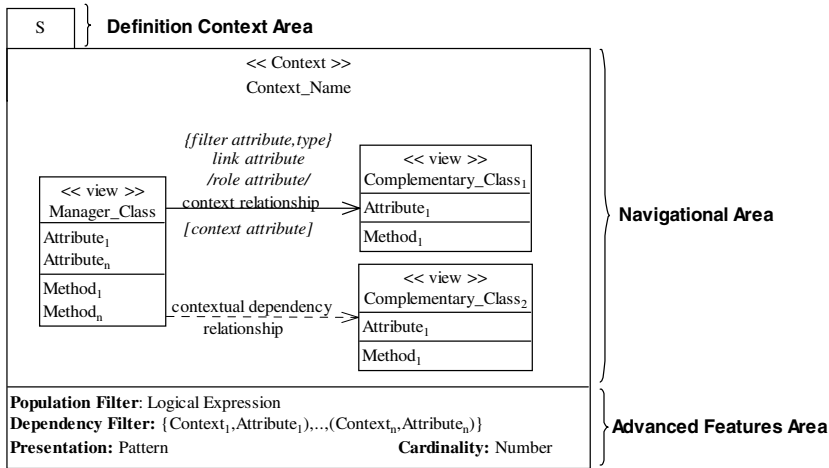


Fig. 1. A Navigational Context

projection over a class of the object model that includes a given subset of its attributes and services.

For each navigational context there is a main class that is called **Manager Class** from where navigation starts. The others classes are called **Complementary Classes** and they contribute to giving additional information to instances of the manager class. These navigational classes contain visible *attributes* and *services* that an agent can see. **Service Links** can be defined for the services of the classes. A service link is associated to a navigational context and it means that the execution of that service will produce a jump to the associated target navigational context.

The navigational classes can be connected by two types of relationships: context and contextual dependency. A **Context Relationship** is a unidirectional binary relationship that can be defined over an existing aggregation or inheritance class relationships. Once again, these aggregation or inheritance relationships come from the Object Model, assuring full compatibility between the navigational model and the other models used in the process of conceptual modeling. This kind of relationship defines an implicit navigational link with a directed navigation between the corresponding classes. Graphically it is represented using solid arrows. In a context relationship, link attributes, context attributes, role attributes and filter attributes can be specified. Now, we introduce these four primitives in detail.

- **Link Attribute:** specifies which visible attribute of the source or target class is involved in the connection defined by the corresponding navigational link.
- **Context Attribute:** specifies the target navigational context of a navigational link.
- **Role Attribute:** indicates the role that makes reference to the relation when two classes have more than one relationship between them. In these cases, the name of the target class cannot be used to identify the relationship unambiguously, and the role attribute provides the solution.

- **Filter Attribute:** indicates a filter on the population of the target class. In this way, the search for specific objects is made easier. It is possible to specify the following basic behaviour for a filter:
 - **Exact (E):** the instances of the manager class in the target context will have the exact value introduced by the user.
 - **Approximated (A):** the instances of the manager class in the target context will have the approximated (like) value introduced by the user.
 - **Range (R):** the instances of the manager class in the target context will be between the limits of the value introduced by the user.

Opposite, in a **Contextual Dependency Relationship** the navigational semantics towards the target class is not defined: this kind of relationship is used to provide the required additional information in the current node, without denoting any further navigation. Thus, no navigational target context is associated. Graphically it is represented using dashed arrows.

3.3 Advanced Features Area

In a navigational context, a **Population Filter** for the manager class can be established during the design. It is represented by a logical expression on the corresponding class attributes. A **Dependency Filter** shows a tuple (context, attribute) for each context ending in the current context. Furthermore, it is possible to specify the **Presentation** style of the information using the following patterns: register mode, tabulate mode and master-detail mode.

Finally, the exploration **Cardinality** allows for the specification of the number of instances of the manager class (with its corresponding dependencies) that will be shown in the context. These features are defined in the lower part of a navigational context (see Figure 1).

Using the primitives presented above, it is possible to specify the semantics attached to the navigation requirements for web-oriented applications. We will show a practical application of all these OOWS primitives in the example coming later. Once the conceptual model has been completed (including the navigational information), a full web-based application can be automatically generated in the next step (Developing the solution). This approach follows the OO-Method approach which reifies the conceptual model into the appropriate target development environment.

4 Case Study

An e-commerce application was used as a case study to validate the OOWS approach. It is an on-line system for ticket sales of a Theatre Company. The requirements for this application are detailed below.

The application interacts with two kinds of users: administrators and internauts. The former are responsible for the maintenance of the information available and the latter are the clients that connect to the Theatre Company homepage through Internet. During the ticket sales session, the internaut indicates the name of the desired show,

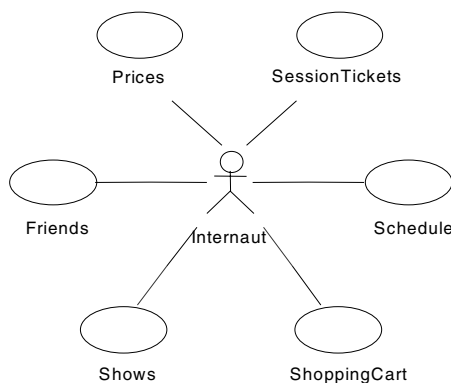


Fig. 2. Use Cases

the corresponding session and the preferred seat locations. Several discounts can be applied (pensioners, students, etc.). These discounts are applied individually on each ticket. Another possibility is to sell season tickets (group of tickets for several shows). Any kind of discount can be applied to season ticket sale. In any ticket sale session, it should be possible to buy any kind of ticket (individual or season, etc.). When a sale is accepted, the seats chosen are marked as occupied. The internaut will have to be identified (SSN, password, name, address and telephone) to be able to collect the tickets from the ticket offices of the company before the show begins. It's not possible to cancel a sale from the Internet. Any cancellation must be made personally at the Ticket Office. The payment is made by a safe transaction using a credit card.

The case study begins with the requirements collection and the conceptual modeling of the **Specifying the System** step. A use case diagram describes the collected requirements. Obviously, to enter into further details of this task is out of the scope of this paper, but in Figure 2, the use cases identified for the Internaut agent are briefly introduced. In the conceptual modeling phase, the following models were constructed: the Object Model, the Dynamic Model, the Functional Model and the Navigational Model.

Before presenting the Navigational Model, we are going to take a brief look at the other three models that constitute the specification of the system functionality. Even taking into account the obvious space restrictions, we consider that it is necessary to have a generic view of the expressiveness related to the more conventional Object, Dynamic and Functional Models. In this way, the need for the complementary Navigational Model introduced in this paper becomes much clearer.

The *Object Model* for this case study is shown in Figure 3. The basic explanation (for reasons of brevity) follows. A Show can have one or more sessions with a Schedule (day, hour). In addition, one or several Companies and also several Artists playing different roles: actors (to_act), directors (to_direct) and authors (Authors) can participate. Optionally, an special Configuration of the Theatre with a distribution of the Seats can be indicated. Each configuration has a price indicated in the TicketItem.

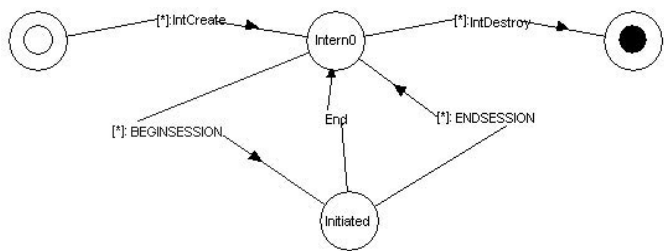


Fig. 4. State Transition Diagram

of the Internaut class when its session is finished was specified. The syntax is shown below:

```
Internaut
Self::(Session = False):IntDestroy()
```

The *Functional Model* was built by capturing the semantics attached in the changes of state. The value of every attribute is modified depending on the action that has been activated, the event arguments involved and the current object state. An example is the Internaut class shown below. Discrete-domain valued attributes take their values from a limited domain.

Attribute:	Session				
Category	Event	Effect	Condition	Current	Value
Discrete-domain	End	=False		True	

To build these previuws models, we used a CASE workbench [14], which supports the OO-Method approach in a unified way.

Finally, the *Navigational Model* was built using the OOWS approach. For this case study, one navigation map was built for the Internaut agent and another for the Administrator agent. Figure 5 shows the navigational map for the Internaut agent with the navigation contexts that have been identified in the specifying the system step.

This map also shows the services that are executed when initiating and finishing a session. When the Web Server receives a request from a new internaut, it executes the IntCreate service to create an instance of the Internaut class, and the service BEGINSESSION to initiate a new session assigning a shopping cart to the internaut. When the system detects that the internaut has finished his session, it will execute the service ENDSESSION to eliminate the shopping cart for unconfirmed purchases and it will execute the IntDestroy service to eliminate the Internaut instance. The dashed arrows indicate that the internaut can access to any one of the exploration contexts. The solid arrows indicate possible navigations between the contexts.

Figure 6 describes the Shows exploration (E) context with information about the offered shows by the Theatre Company. The navigation begins in the Show class that acts as the manager class. Title, Picture, BeginDay and EndDay of a show will be shown.

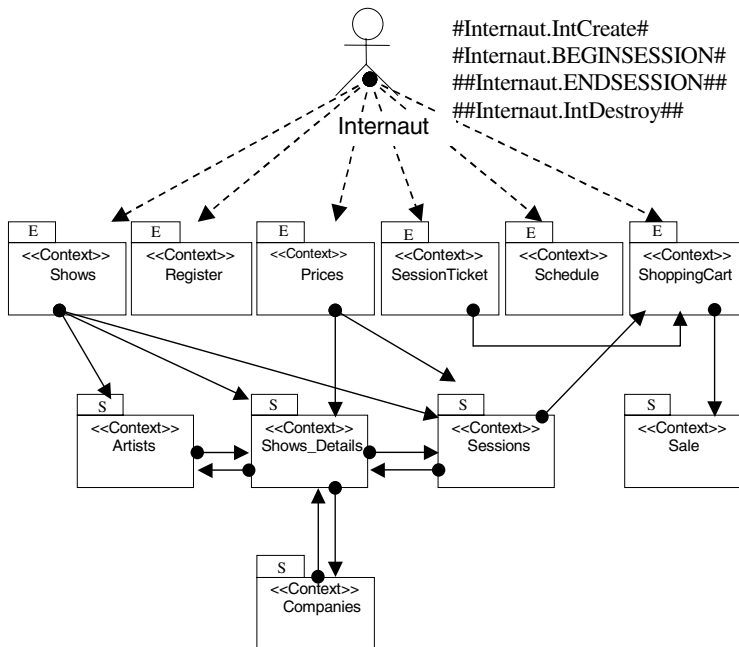


Fig. 5. Navigational Map (Internaut Agent)

The Artists class is a complementary class of this context. In addition, the contextual relationship (represented by solid arrows) between Show and Artist will also present the artist name (ArtName) This artist plays the role to_direct in the Object Model (see Figure 3).

This contextual relationship also defines a navigation from "Shows context" to "Artists context" (indicated by the context attribute [Artists]). An internaut would access this target context using the artist name context attribute (to_direct.ArtName). Furthermore, the Show class has two reflexive contextual relationships. One relationship allows users jumping to the "Shows_Details context" using the Picture or Title attribute of a Show. And the other relationship allows jumping to the "Sessions context" using the BeginDay or EndDay attribute of a Show. All these navigations can be seen in the navigational map (Figure 5). The population of this class will appear in a register mode (see in advanced features area). The other contexts for the case study were described using a similar approach.

In the **Developing the Solution** step, a strategy for component generation for the automatic implementation of the system was defined. This strategy proposes a structure which is composed of three layers: Data, Business Logic and Presentation.

The *Data* layer is represented by the relational database that stores the population of each class that has been identified during the design. The SGBD used was SQL-Server version 7.

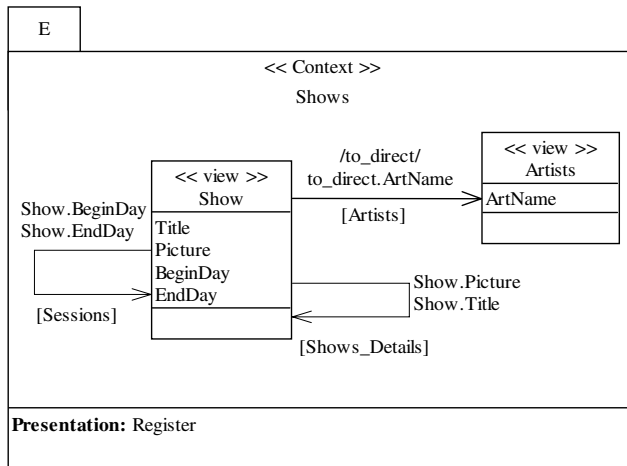


Fig. 6. Shows Context

The *Business Logic* layer contains the business logic for each class specified in the conceptual model. In this layer, we also define a persistence class to retrieve/update from/to the database the instances of the classes that will be used by the application, creating and deleting instances and other auxiliary ones. The Remote Data Objects (RDO) technology with ODBC has been used to perform the communication with the Data layer.

We defined a component (ActiveX/COM) for each OO-Method class that implements its interface in the object model. A service manager is defined to begin the execution of these services, including transaction management, integrity constraints and trigger verification. Any service execution is characterized by the following sequence of actions:

1. Object identification: the object acting as server has to be identified. This is an implicit condition for any service, except if we are dealing with a new event. Their values are retrieved at this moment.
2. Introduction of event arguments: the rest of the arguments of the event being activated must be introduced.
3. State transition correctness: we have to verify in the STD that a valid state transition exists for the selected service in the current object state.
4. Precondition satisfaction: the precondition associated to the service must hold. If not, an exception will arise to notify that's its precondition has been violated.
5. Valuation fulfilment: once the precondition has been verified, the induced event modifications are performed in the selected persistent object system.
6. Integrity constraint checking: to assure that the service activation leads the object to a valid state, we must verify that the integrity constraints hold in this final resulting state.

7. Trigger relationships test: after a valid change of state, and as a final action, the set of condition-action rules that represents the internal system activity has to be verified. If any of rules hold, the corresponding service activation will be triggered.

The *Presentation* part contains all the presentation logic of the application. In this case study, the chosen architecture to implement this part is: Visual Basic version 6.0 as implementation language and the Internet Information Server with technology ASP (IIS/ASP) as web server. A WebClass is the main component of an IIS/VB application. It is a component that is called from an ASP page and provides the developer a way of responding to requests from Web clients. The WebClass is composed by WebItems that returns HTML to the client browser. Thus, we defined a mapping from OOWS primitives into an IIS/VB project. In this mapping the following patterns are defined:

- **Navigational Context:** a WebItem is created with an event that is executed by default for each navigational context. This event contains the necessary code to show all the elements of this context. A link on the left side of the page is included for the exploration contexts.
- **Navigational Classes:** they are instances of the classes implemented in the business logic layer of the application. It is necessary to declare a manager class with all the instances that fulfill the defined filters. The complementary classes are obtained from the navigation functions by using the corresponding roles defined in the business logic layer. The attribute values of the classes are obtained from the properties that have been defined in the business logic layer.
- **Filters:** the dependency and context filters are used to retrieve the population of the manager class. Its use provides the view of the objects which are actually alive in the system. The filters of the complementary classes are used to retrieve the instances of the class using the corresponding navigation functions.
- **Context Relationship:** if the context relationship has a link attribute, a link for each such attribute will have to appear in the WebItem that has been created for the target context. Instead, if the context relationship does not have a link attribute, a link will have to appear in the WebItem that shows the alias of the target context. This link does not have parameters because it will not have any dependency filter attached to it. When a filter attribute appears in a context relationship, the required information in the origin context will depend on the filter type (exact, approximated and range). The parameters passed into the method call of the target context include the filter result. This filter will have to be executed in the target context when the population of the manager class is retrieved.
- **Contextual Dependency Relationship and Navigational Links:** It not is necessary to define patterns for the mapping of relationships of this type since it only helps to represent information in the diagram, but does not have any associated functionality.
- **Services:** a customized event of the WebItem that represents the context is created for each service that can be executed from the current context. This event will call the service manager that really executes the services.
- **Service Links:** if a service can obtain all the arguments at the current page, the service link will call the customized event that will execute the service. On the



Fig. 7. Shows Context

other hand, if a service cannot obtain all the arguments at the current page, it will load a page. This page contains a form, textboxes for the attribute entries, and a button labeled ACCEPT that will call the customized event to execute the service.

- **Read/Write Attributes:** are mapped in textboxes that show the value that the corresponding field of the current record contains. We introduce a form for each record that allows for sending those values to the customized event that will gather the arguments and will execute the service.
- **Presentation Pattern:** allows for the presentation of the information associated with each navigational context. A "template" is specified for each type of presentation (register mode, tabulate mode and tabulate master-detail mode).

Detailed information about these patterns can be found in [13]. The obtained result with the described mapping is a web-based application. Figure 7 shows an example of an HTML page generated by this mapping. It presents the Shows page of the context described in figure 6. The menu on the left side indicates the exploration contexts.

5 Results and Conclusions

In this paper, we have presented the OOWS approach for web-based applications development. This is based on the definition of primitives to capture navigational requirements. Currently, we are applying this approach to several real-world e-commerce applications including the Theatre Company that was presented here. The experience

gained has allowed us to put into practice the OOWS approach and to refine the proposed primitives. Also, a strategy for the component generation oriented to an automatic implementation of the system was defined. It have been proposed implementation patterns for an architecture using Visual BASIC and ASP. However, this patterns can be also defined for other platforms. The work in progress involves the extension of the modeling language to specify security features and integrity validation of the navigational model with respect to the other elements of the conceptual model. Nowadays, we are working on a translator/code generator based on XML [7] and we are defining the patterns for generating client interfaces into multi-device channels (HTML, XML, WAP, etc.).

References

1. Bonifati A., Ceri S., Fraternali P., and et al. Building multi-device, content-centric applications using webml and the w3i3 tool suite. In *19th International Conference on Conceptual Modeling (ER'00)*, Salt Lake City, USA, 2000. Springer-Verlag.
2. Lange D. An object-oriented design method for hypermedia information systems. In *Hawaii International Conference on System Science*, 1994.
3. Schwabe D. and Rossi G. The object-oriented hypermedia design model. In *Communications of the ACM*, pages 45–46, 38(8) 1995.
4. Booch G., Jacobson I., and Rumbaugh J. *The UML Language Users Guide*. Addison-Wesley, 1999.
5. Mecca G., Merialdo P., Atzeni P., and Crescenzi V. The araneus guide to web-site development. Technical report, University of Roma, Roma, Italy, 1999.
6. F. Garzotto, Paolini P., and Schwabe D. Hdm - a model-based approach to hypertext application design. In *ACM Transactions on Information Systems*, pages 1–26, 11(1) 1993.
7. <http://www.w3.org/TR/1998/REC-xml-19980210>. Extensible markup language (xml) 1.0, 1998.
8. Jacobson I., Christerson M., Jonsson P., and Overgaard G. *Object Oriented Software Engineering, a Use Case Driven Approach*. Addison -Wesley, Reading, Massachusetts, 1992.
9. Gomez J., Cachero C., and Pastor O. Extending a conceptual modeling approach to web application design. In *Conference on Advanced Information Systems Engineering (CAiSE'00)*, pages 79–93. Springer- Verlag, 2000. LNCS 1789.
10. Baresi L., Garzotto F., and Paolini P. From web sites to web applications: New issues for conceptual modeling. In *Workshop on Conceptual Modeling and the Web (ER'00)*. Springer-Verlag, 2000. LNCS 1921.
11. Pastor O., Insfrán E., Pelechano V., Romero J., and Merseguer J. OO-Method: An oo software production environment combining conventional and formal methods. In *9th Conference on Advanced Information Systems Engineering (CAiSE'97)*, pages 145–159, Barcelona, Spain, June 1997. Springer-Verlag. LNCS (1250).
12. Pastor O., Fons J. J., Abrahao S. M., and Ramon S. Object-oriented conceptual models for web applications. In *4th Iberoamerican Workshop on Requirements Engineering and Software Environments (IDEAS'2001)*, San Juan, Costa Rica, 2001. (in Spanish).
13. Pastor O., Molina P. J., and Aparicio A. Specifying interface properties in object oriented conceptual models. In *Working Conference on Advanced Visual Interfaces*, pages 302–304, Italy, 2000. ACM Press.
14. Pastor O., Pelechano V., Insfrán E., and Gómez J. From object oriented conceptual modeling to automated programming in java. In *17th International Conference on Conceptual Modeling (ER'98)*, pages 183–196, Singapore, November 1998. Springer-Verlag. LNCS (1507).

Tools for the Design of User Friendly Web Applications^{*}

Nieves R. Brisaboa, Miguel R. Penabad, Ángeles S. Places, and
Francisco J. Rodríguez

Departamento de Computación, Universidade da Coruña, 15071 A Coruña, Spain.
brisaboa@udc.es, {penabad,asplaces,franjrm}@mail2.udc.es

Abstract. The usability of Web pages and applications is a fundamental factor in their success. Web designers are taking advantage of the new technologies (scripting languages, Java, etc.) to increase the power, user friendliness and easiness of use of Web pages.

This work presents two techniques that can be used to achieve this goal: the use of cognitive metaphors to build user interfaces, and the use of Bounded Natural Language, to allow the user to express queries in an intuitive way.

1 Introduction

Internet is now becoming the most valuable vehicle to carry out tasks as different as finding information or perform commercial transactions. There are several reasons for this fact. Among them, there are technological reasons (the technology improves at a really fast pace) and economical ones, finding that the technology is now cheaper. However, we believe that there is one specially important reason for the increasement of use of Internet: the growing easiness of use of the interactive Web pages, that can nowadays be considered real computer applications, due to their enhancements with scripting languages, interoperability with databases, and specially programming languages like Java [5].

The usability of the user interface is a crucial aspect for the success of a Web page, because if the users find it difficult, they will not use it. A good example of this fact is the evolution suffered by the search engines (see, for example, [14]): initially, they offered just an “edit box” where the user entered some words that specified the subject of her search. Most of the current search engines provide now a much more friendly interface, with some natural language interfaces, use of folders to clasify the answers, and some other innovations. Therefore, it seems clear that the design of interactive Web pages must consider the use of a methodology to design their user interface, in order to produce a functional, powerful and easy to use interface. This paper presents two different ideas that can be used to build user interfaces improving their easiness of use. These ideas are briefly defined as:

^{*} This work was partially supported by CICYT grant TEL99-0335-C04-02 and Xunta de Galicia grant PGIDT99PXI10502A.

- Use of *cognitive metaphors*, that is, build the Web pages using escenarios that represent the real world and are similar, both in aspect and functionalities, in the Web page and in the real world.
- Use of *Bounded Natural Language* [6]. It is a technique that consist on offering the user a set of sentences (in natural language) with gaps. The user must choose the sentences she is interested in, and fill the gaps. Finally, the set of selected sentences with the gaps filled will represent the user query.

The use of these techniques will be explained in this paper, along with the advantages they present. Nevertheless, we must note that they do not conform a methodology of user interface design, but they can be used to efficiently improve the aspect and functionality of any Web interface.

The rest of this work is organized as follows. Section 2 briefly describes our project, a Virtual Library for Emblem books, as well as the Spanish Emblem Literature. Section 3 describes the use of cognitive metaphors and their use in our project. Section 4 describes the Bounded Natural Language as a powerful technique to build easy to use query interfaces. The query system of the project described in this paper is shown in Section 5. Finally, Section 6 shows our conclusions.

2 Example of Use: Virtual Library of Spanish Emblem Literature

Emblem Literature became very popular all around Europe during the 16th and 17th centuries. This literature is a very rich and complex source of information related to the customs of those centuries in Europe. It provides data about society, morality, customs, knowledge and conventions. It is basically a moral literature directed at the education of royalty and the bourgeoisie, although there are also works directed at women, the clergy, etc.

Each emblem was illustrated with a picture, having a phrase or motto (usually in Latin), a little poem (epigram) explaining the moral idea and a commentary on the idea, usually citing or quoting classical writers or the Holy Scriptures and using analogies.

Given their characteristics, emblems are very useful to a wide range of researchers from a variety of disciplines such as History of Literature, History of Art, Anthropology, Sociology, or Philology. Despite its beauty and usefulness, it is not expected that the books of emblems will be re-edited at present, and the original editions are scarce and not easily accessible, usually in libraries with historical archives and in most cases are only to be found on microfilm. Consequently, researchers are denied the possibility of easy access to a literature, which like no other represents the diverse nature of Baroque culture.

The main idea of our project was to build a Virtual Library of Spanish Emblem Literature. The underlying database not only stores the digitizations of the emblems but also the results of their analysis by specialists in Philology, Latin, Literature and Art. This project offers the possibility to access all the

information contained in the emblem books and consequently avoids unnecessary travels or the investment of large amounts of money in microfiche or microfilms. It will be possible, simply by having a computer and a modem or by going to an institution like a Library or University connected to Internet, to select the books or emblems of interest and consult them. Researchers of many different fields (e.g., linguists, anthropologists, and historicists) will consider this application an invaluable help in their research.

Following the ideas described in this paper, we have built a Virtual Library of Spanish Emblem Literature¹. It shows how the use of metaphors and Bounded Natural Language helped us to build a user friendly and powerful user interface.

3 Use of Metaphors

The use of metaphors or analogies is a widely studied technique in the Human Computer Interaction research field [8]. It is based in the use of something known to the user that is translated to a new domain. In some cases, it based on the similarity of the physical aspect; in others, on the similarity of goals or tasks that are performed. Early examples are word processors, using the metaphor of a typewriter, or a database that uses a filing cabinet metaphor. Is is clear that the first one is based on the aspect similarity (the user sees a sheet of paper where he or she writes), while the second one is based on the similarity of the goals achieved by both database and filing cabinets, such as storing and retrieving information.

One of the most successful metaphors in the Web is the well known “shopping cart” metaphor, used on the majority –if not all– the online shops. The success of the online shops, and therefore the success of the shopping cart metaphor, is a good indication of the advantages of using a good metaphor.

3.1 The Library Metaphor

The Web page of the Virtual Library presented in this work, uses a library metaphor. This page is similar (in this case, both in its aspect and its functionalities) to a real library. Any user can access the different services offered by this virtual library by “going” to the appropriate place:

- Register (“Inscripción”): Any user must become a member of the virtual library to access some of the offered services. The registration is carried out, as in a real library, at the front desk. The information given in the registration process, except for the login name, can be marked public or private. This information includes full name and position, address, telephone and fax numbers, e-mail, etc.
- Find information about other members (“Información de socios”): Registered users can access the information about other members that was marked as public. There appears a new cognitive metaphor: the use of the cards with

¹ Available at <http://emilia.dc.fi.udc.es/CICYT96/index.html>.

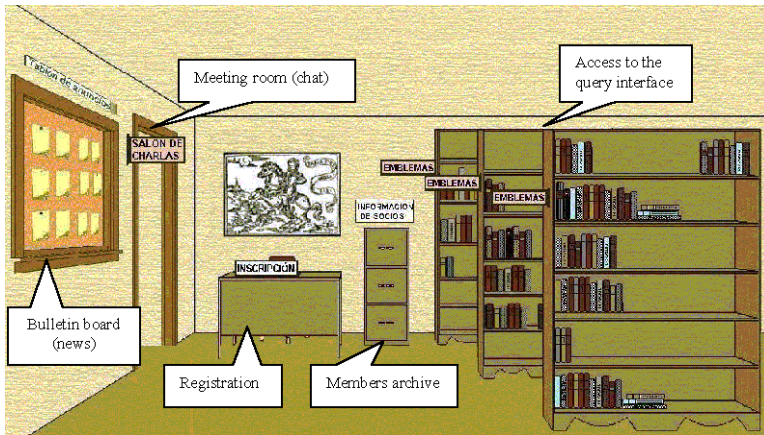


Fig. 1. Main page

the information, and markers with the initials of the members names to quickly browse through the cards.

- Use the meeting room (“Salón de charlas”): Registered users can meet and discuss any kind of information by entering the meeting room (using the door on the left of the library). This service is implemented as a common chat server, and the Web interface is shown in Figure 2. It shows a meeting room where conversations take place. To exit the meeting room and go back to the main entrance of the virtual library, the user must go again through the door.
- Use the bulletin board (“Tablón de anuncios”): The purpose of the bulleting board is to share information among the members of the library, or ask questions and receive answers. The interface shows a bulletin board with stacks of notes. Each stack refers to the same subject, which is the only information shown on them. Clicking on a stack, the full text of the notes is shown, and there is the possibility to answer them, adding a new note to the stack, or put a new announcement in a new stack.
- Search and browse through the emblem books: By clicking on the shelves of books, the user accesses the query system. It uses a book metaphor to express the queries and to show the results. One of the main characteristics of the query system is the use of Bounded Natural Language, shown in the next section.

4 Bounded Natural Language

As stated in the introduction, the Web interfaces used to search for information have evolved from simple edit boxes to much more powerful and intuitive techniques. Some search engines offer the possibility of introducing a question that

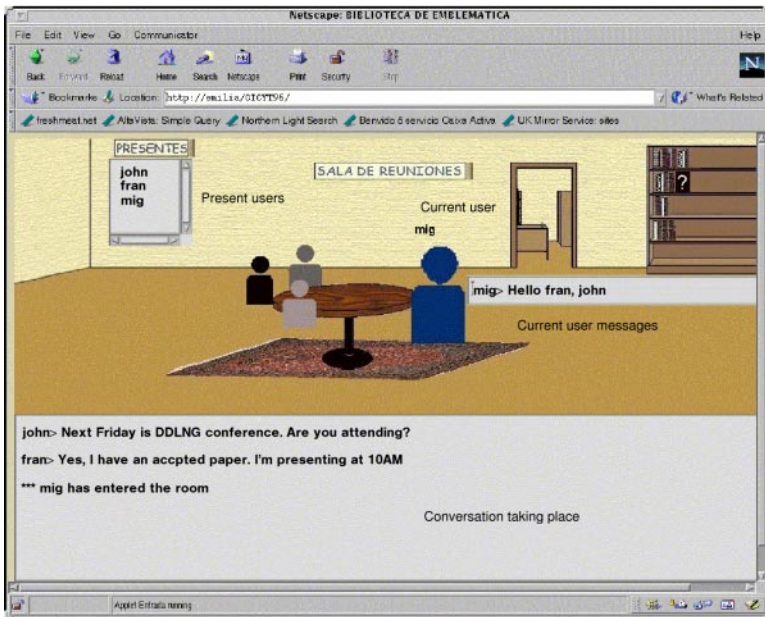


Fig. 2. Meeting room

is solved by the system by showing a list of Web pages that –supposedly– can help the user to find the answer to the question. That is, the interface is based on the use of Natural Language to define a query.

The use of Natural Language is, undoubtedly, very powerful. However, it has many inconvenientes, because of the intrinsic nature of the Natural Language. Languages do not follow a strict grammar (like programming languages), they contain synonyms, phrases or idioms, and their sentences can be ambiguous. Therefore, it is very difficult to find a technique that accurately transforms a natural language sentence into a formal query language to interrogate the database where the information is stored.

For example, using a search engine (in this case, <http://aj.com>) and asking “*What is the distance between Madrid and Barcelona?*”, the server provides some answers or redirects us to some questions pre-stored in the database with their answers. Some of them are correct and lead us to the solution for the question. However, others are clearly not. One of those was “*Where can I find a metric conversion table?*”, probably obtained by the fact of having the word “distance” in the query.

It seems clear that the use of pure natural language to build Web interfaces presents many problems with respect to its management and the results produced. Our proposal is to use a different technique, what we call Bounded Natural Language.

Basically, a Bounded Natural Language sentence is a sentence in natural language with some gaps that the user must fill in order to express the search conditions. Usually, several sentences about the same topic are presented to the user at the same time. The user must choose which sentences will be used, and fill their gaps. Finally, the set of selected sentences with their gaps filled will express, in natural language, the whole query the user is asking. The following example shows a set of Bounded Natural Language sentences before and after the gaps are filled (both sentences are selected to express the query).

Before: *“I am interested on emblems defined by [Some of/All/At Least ... of] the following topics:[...]. I am [Specially/Only/Also] interested in those emblems that quote [Some of/All/At Least ... of] the following authorities:[...]”*

After: *“I am interested on emblems defined by [Some of] the following topics:[sin; virtue; prince; king]. I am [Also] interested in those emblems that quote [At Least 3 of] the following authorities: [Holly Scriptures; Alciato; Graciano; Alfonso X]”*

The example shows the two main types of gaps that are used in a Bounded Natural Language sentence:

- *Condition gaps:* Used to restrict the data that will be retrieved, these gaps are to be filled with words by the user. Usually the user will be able to type a list of words (separated by semicolons) but, if the number of possible words for a given gap is short and/or the words are restricted (like the selection of authorities in the previous example), the words can be chosen from a list.
- *Modifiers:* These gaps are to be filled with just one value, and they are used with two different objectives:

- Characterize a condition gap: These modifiers, like *[All/Some of/At Least ... of]*, usually precede a condition gap and establish whether all the words typed in the condition gap must be considered (choosing *All*), only some of them (choosing *Some of*), or they establish the minimum percentage or number of the words that must be considered (choosing *At Least ... of*).

The first option, demanding that the documents must contain all the typed words, does not offer any doubt about how it works. Similarly, demanding that the documents must contain a minimum number or percentage of words (selecting, for example, “At Least 3 of”, or “At Least 75% of”) is also clear. However, choosing “Some of”, the number or percentage of words that must be present in the documents is not fixed. This percentage will depend on the relevance of the sentence that includes this modifier (this will be explained more clearly in the description of the next type of modifier).

- Give different relevance to a sentence. Modifiers like *[Only/Especiallly/Also]* give a different relevance to a sentence with respect to the others. The “weight” of the relevance for this example would be higher using *Only* (the conditions expressed by the

sentence must be satisfied by any document to be retrieved) than using *Epecially*. Choosing *Also*, it acts like a logical OR.

The modifiers used to give a different relevance to a sentence also change the behavior of the modifiers used to characterize a condition gap, when options like “Some of” are chosen. This change of behavior is reflected in the variation of the percentage of words that the documents must contain. Let us look at the sentences in the previous example:

“I am interested on emblems defined by [Some of] the following topics:[sin; virtue; prince; king]. I am [Also] interested in those emblems that quote [At Least 3 of] the following authorities: [Holly Scriptures; Alciato; Graciano; Alfonso X]”

The relevance of the second sentence is similar to the first one, because the chosen modifier was “Also”. The modifier in the first sentence, “Some of” is translated into a percentage of words that the documents must include. For example, let us assume this percentage is 75%. However, consider these sentences written as follows.

“I am interested on emblems defined by [Some of] the following topics:[sin; virtue; prince; king]. I am [Specially] interested in those emblems that quote [At Least 3 of] the following authorities: [Holly Scriptures; Alciato; Graciano; Alfonso X]”

It is clear that now the second sentence is more important than the first one, because the user is “specially” interested in the quotations of the documents. Therefore, the relevance of the first sentence decreases, and so does the necessary percentage of words that describes the topic of the emblems. For example, this percentage could be now defined as 50%. This behavior becomes more complex when the number of the sentences offered to the user and the number of modifiers increase.

The following section shows the query system of the Virtual Library. It is a combination of the use of cognitive metaphors (library and book metaphors), which are used to build simple queries and present the results, and the use of Bounded Natural Language described in this section, used to build more complex queries.

5 The Query System

By clicking on the shelves in the library shown in Figure 1, the user accesses the query system of the virtual library. The main entrance allows the user to search for a specific document (Emblem) or browse through the whole set of documents, as Figure 2 shows. After the user enters the query system, she must choose either a book or author, or all the emblem books. Then, she can browse through the selected list of books, or restrict the search.

In order to perform a simple search, the image shows an open emblem book, where each part is located at the same place as in a real emblem: the motto and the image, the poem, and the comentary explaining the emblem. Therefore, the

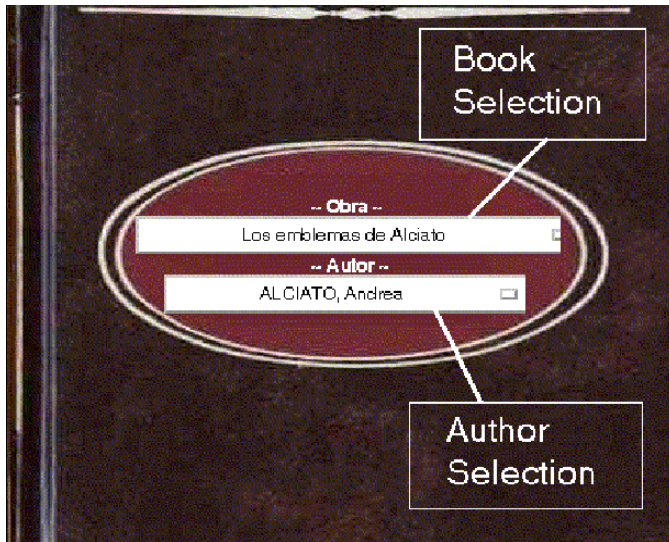


Fig. 3. Selecting a book or author

user intuitively knows where (physically on the pages of the book) to establish the desired search restrictions.

This interface permits to establish simple search conditions, such as type of verse, words included in the motto, etc. However, there is the possibility of using more complex search conditions, that involve structured as well as non-structured data [23,7]. The search conditions are expressed by using Bounded Natural Language sentences (there are 5 sentences in this interface), as shown in Figure 4 (the interface is at this moment only in Spanish because it is mainly used by Spanish-speaking researchers). Each sentence must be “enabled” (using the checkbox on the left) to be taken into account. Then, the gaps of the selected sentences must be filled. The presence of the modifier gaps (Especially/Only) at the beginning of the sentence are used to alter the relevance of each sentence. Note the presence of a list box associated to the query gaps. These lists are included because the words that can be typed in these gaps are limited and known.

After the gaps are filled, the user can read the sentences in order to check if they express what she wants to retrieve. It is a simple task, because the user will read some natural language sentences and quickly understand their meaning, opposed to the majority of interfaces that offer boolean search capabilities, where understanding the meaning of the search conditions is not immediate.

Once the system retrieves the documents that match the query condition, they are shown using again a book metaphor. The user can select a book by clicking on it. Then, the information about the book is shown, and the user can browse through the pages and see the data of the emblem in a layout that resembles an original emblem. There are no buttons, but only “sensible” parts of a book.

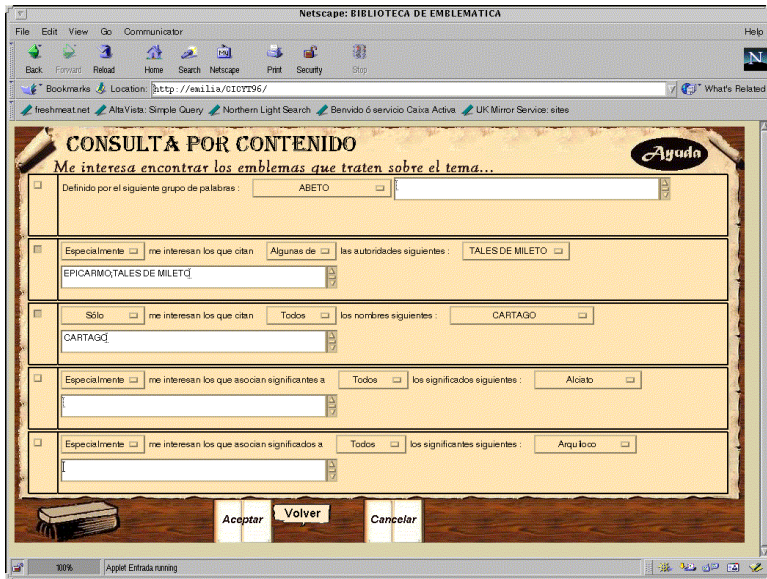


Fig. 4. Complex search using BNL

Of course, there are functionalities in this interface that do not come from the analogy of the book (because the book does not have them), such as the possibility of seeing the digitized original page by clicking on the image. However, these new functionalities are well integrated in the design and they should not introduce any difficulty to the users.

6 Conclusions

We have presented two techniques used to build Web applications that are powerful while being user friendly and easy to use. The use of cognitive metaphors in every possible aspect of the interface makes it intuitive. Users know how to work with it because they know how to work with the original elements, in this case a library and a book. Moreover, the use of Bounded Natural Language to express queries is a powerful technique that allows users to express queries without having to learn any query language, again in an intuitive manner.

The interface presented in this work is the result of the evolution of different prototypes along the years. Even without making any formal experiment with users, their comments about their growing satisfaction lead us to believe that the use of the techniques presented in this work can be used to build powerful and easy to use Web interfaces.

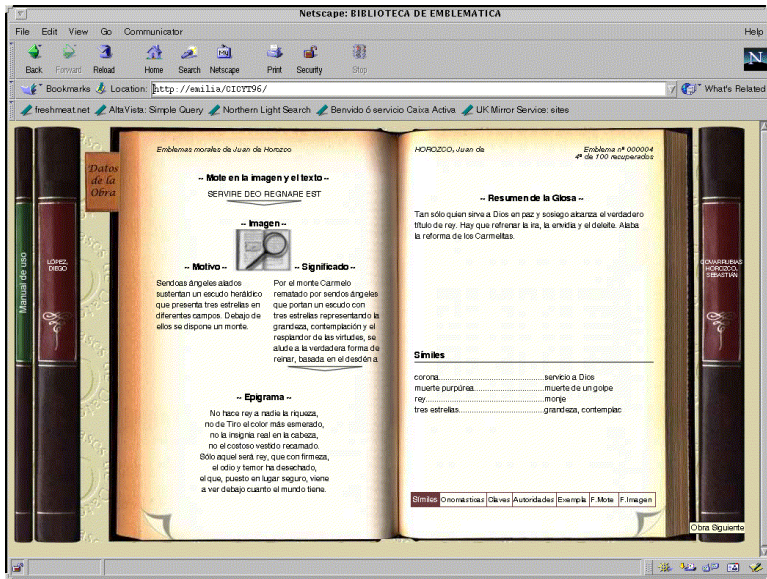


Fig. 5. A virtual book

References

1. AQUARELLE: The Information Network on the Cultural Heritage. CCLRC-RAL (UK), CNR-CNUCE (Italia), CNR-ITIM (Italia). ICS-FORTH (Grecia), IMAG (Francia), INRIA (Francia), LIRMM (Francia). <http://aqua.inria.fr/Aquarelle>.
2. Abiteboul, S., Buneman, P., Suciu, D. *Data on the Web. From Relations to Semistructured Data and XML*, Morgan Kaufmann Publishers, 2000.
3. Baeza-Yates, R.; Ribeiro-Neto, B. *Modern Information Retrieval*, Addison-Wesley, 1999.
4. J. Davis. *Creating a Networked Computer Science Technical Report Library*. D-Lib Magazine, Sept. 1995.
5. <http://www.javasoft.com>.
6. Penabad, M. R., Durán, M. J., Lalín, C., López, J. R., Paramá, J. R., Places A. S., and Brisaboa, N. R. (1999). Using Bounded Natural Language to Query Databases on the Web. In *Proceedings of the 5th International Conference on Information Systems Analysis and Synthesis: SCI'99 and ISAS'99*. Volume 2, pp. 518-522. 1999, Orlando, Florida.
7. Salton, G. *Automatic information Organization and Retrieval*. McGraw-hill, 1968.
8. Zetie, C. *Practical user interface design: Making guis work*. McGraw Hill, 1995.

EProMS: An E-commerce Based Process Model for Cooperative Software Development in Small Organisations

Awais Rashid¹, Ruzanna Chitchyan², Andreas Speck³, and Elke Pulvermueller⁴

¹Computing Department, Lancaster University, Lancaster LA1 4YR, UK
marash@comp.lancs.ac.uk

²Computing for Commerce and Industry, The Open University, UK
r.chitchyan@lancaster.ac.uk

³Wilhelm-Schickard Institute for Computer Science, University of Tübingen, 72076
Tübingen, Germany
speck@informatik.uni-tuebingen.de

⁴Institut fuer Programmstrukturen und Datenorganisation, University of Karlsruhe, 76128
Karlsruhe, Germany
pulvermueller@acm.org

Abstract. Existing process models for software development are too resource intensive for small organisations. This paper proposes EProMS, an e-commerce based process model which takes into account the limited resources of these organisations. It distributes the software development process over several small organisations. These organisations, then, co-operate through an e-commerce layer to produce market-driven software. On competitive markets the distribution allows organisations in the *co-operation* to specialise in specific aspects of software development. This helps them to reduce costs and time to market and achieve a high degree of users' satisfaction.

1. Introduction

Small organisations comprise a major portion of the software industry. A small organisation can be defined as an organisation whose limited resources constrain it from following the complete software development cycle. This paper proposes a process model for software development in these organisations. A process model identifies the various phases or activities in a process without ascending on how to carry out these activities. To date several process models for software development have been proposed. A major criticism of classical models is that they are too resource intensive to be adopted by small organisations. Cost-effectiveness, reduced time to market and users' satisfaction play a crucial role for small organisations. When using existing models small organisations have to accept a trade-off between a high degree of users' satisfaction and reduced costs and time to market. The large number of phases/activities in existing models require availability of vast resources (such as time, human, financial, etc.) which are available in big corporations. Small organisations have limited resources and tend to cut costs and time to market¹ through bypassing some of the phases. Consequently certain features/requirements get ignored reflecting on software quality and, hence, users' satisfaction.

¹ This is particularly crucial when competing against bigger corporations capable of swamping the market.

The proposed **E-commerce based Process Model for Small organisations (EProMS)** takes into account the limited resources of small organisations and, in contrast to its existing peers, focuses on *co-operation*. EProMS distributes the software development process over several small organisations. These organisations, then, co-operate through an e-commerce layer to produce market-driven software. The distribution allows organisations in the *co-operation* to specialise in specific aspects of software development. This in turn provides an organisation the opportunity to capitalise on its advantages (experience, technical knowledge, etc.) and cut costs. Specialisation also helps reduce the time to market by achieving significant parallelism of software development activities and eliminating the need for an organisation to switch between these activities. Furthermore, it leads to an improvement in product quality which implies higher users' satisfaction. Being a process model EProMS describes the activities of various organisations within the *co-operation*. It does not describe how each organisation carries out its specific activities as these depend on variables local to the organisation.

The next section discusses the major existing process models for software development and their shortcomings when used in small organisations in general and those producing market-driven software in particular. This is followed by a description of EProMS and its advantages. Various possible barriers in the adoption of the model are also identified. The final section summarises and concludes the paper.

2. Existing Process Models

A number of process models have been proposed for software development. Of these the waterfall model [14] was one of the first. The waterfall model divides the software development process into sequential phases². Each phase takes as input a set of deliverables and produces another set of deliverables as its output. Although the model makes the software development process more visible the large number of phases involved are too resource intensive to be followed by small organisations. For example, the various deliverables consume a considerable portion of human resources and equipment in such an organisation. The systems are introduced with a "big bang" resulting in objections/reservations by the users [3]. Also, the sequential nature of the model renders it insufficient for organisations competing for market-driven software; the various stages and iterations between them are severely time consuming in such a competitive scenario. The problem is more serious in situations where small organisations are competing against bigger corporations as high expenses and delivery delays threaten the existence of the small organisation. Although variants of the waterfall model have been proposed e.g. [12], these aim at capturing the activities embedded within the various phases and do not focus on small organisations.

The major alternative to the waterfall model has been the spiral model [1] which is based on assessment of management risk items at regular stages in the project. Before each cycle a risk analysis is initiated and at the end of each cycle a review procedure assesses whether to move onto the next cycle of the spiral. Where the model counters the various risks in a timely fashion, it introduces an additional layer of resource consuming activities rendering it unusable in organisations with limited resources.

² In the adapted waterfall model the various phases are not totally sequential as iteration between phases is possible though expensive.

Furthermore, the model, like the waterfall model, is sequential in nature and there is an additional delay due to the risk analysis involved. This results in valuable time being lost in releasing a market-driven product. The spiral model, therefore, suffers drawbacks similar to the waterfall model when adopted by small organisations.

Other process models such as rapid prototyping [8, 9] and incremental development [4] aim at constructing a partial system and working from there onwards. These models are considered to be more suitable for small organisations. However, the low visibility of users' needs leads to an open ended process requiring backing by large resources such as those available in bigger corporations. Also, the large number of potential iterations render the models too expensive for small organisations both in terms of resources and time. Where prototyping can help understand the user needs the iterations can consume considerable time for a competitor to capture the mass-market.

More recent approaches such as the macro and micro process of G. Booch [2] and the Rational Unified Process [10] describe how to carry out the various activities/phases identified by a process model. Since they do not fall within our definition of a process model they are beyond the scope of this discussion. The same applies to more recent evolutionary approaches such as STEPS [7]. Although exceptionally suitable for highly motivated small teams STEPS does not aim at providing a process model. Besides it focuses on small teams within large organisations. Small organisations are not considered specifically.

Another drawback of the models discussed above is that they require teams of specialists to carry out the various phases. Employing a large number of people is not feasible for small organisations. Although the same team could specialise in more than one phase it is normally not the case as the advantage gained by carrying out tasks in parallel and having various phases interleaved or overlapped are lost.

Another shortcoming is the cultural shock to the new software engineer (joining a small organisation) due to the lack of use of the process models s/he has learnt. Not that small organisations are not committed to good software engineering practices, the proposed practices are too resource intensive for them.

3. EProMS

EProMS is a process model supporting software development in small organisations. It focuses on:

- co-operation between small organisations
 - specialisation of small organisations
- and leads to reduction in costs and time to market and a high degree of users' satisfaction.

EProMS proposes formulation of an e-commerce based *co-operation* between small organisations to produce market-driven software (cf. fig. 1(a)). As shown in fig. 1(b) each organisation in the co-operation concentrates on a niche in the software development process. This allows the organisation to specialise in a particular aspect of software development and to devote its resources to this specific aspect. Consequently there is an increase in efficiency as each firm can develop and exploit firm specific competitive advantages such as trained and skilful employees, fully utilised equipment, etc. Specialisation within the *co-operation* also provides an

opportunity to build up stable relationships among the members. At the same time each business fully maintains its independence and flexibility.

The *co-operation* in fig. 1(b) includes (but is not necessarily limited to) the following organisations:

- Market Study Companies: specialising in studies of people and organisations
- Designer Companies: specialising in component design
- Component Developer Companies: specialising in component development
- Component Certification Body: specialising in certification of developed components
- Component Assembler Companies: specialising in assembling products from components and sales and marketing of these products

It should be noted that the *co-operation* shown in fig. 1(b) is an example *co-operation*. There can be other variations of a *co-operation* e.g. having another company specialising in domain engineering.

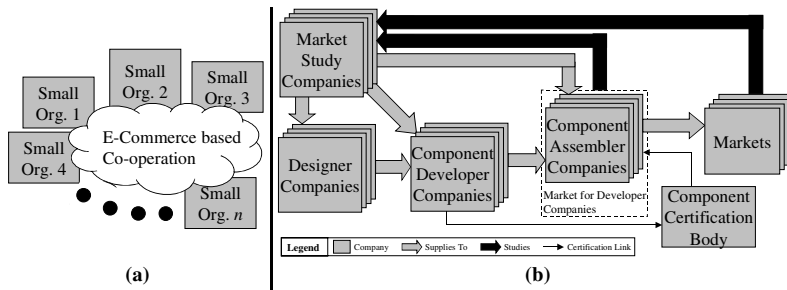


Fig. 1. (a) An e-commerce based *co-operation* among small organisations **(b)** An example *co-operation*

The market study companies in the example *co-operation* (fig. 1(b)) specialise in studies of people and organisations. A market study company can employ a combination of ethnographers and technologists who monitor and study the market in order to identify needs for new products or changes/enhancements to existing products. They also study the component assembler companies (market for component developer companies) in order to identify needs for new components to build new products or change/enhance existing products. The assembler companies purchase studies identifying new products required or changes/enhancements required to existing products. An assembler company can also request the market study company to measure the response of the markets to released products.

The designer companies specialise in designing components for a particular domain or set of domains and purchase studies identifying needs for new components in their area of specialisation. It should be noted that the need for new components can be identified as a direct consequence of an assembler company purchasing studies identifying a potential product. Since the *co-operation* suggests sharing ethnographic studies with software designers it is worth mentioning that work has been carried out for such an exchange e.g. [15].

The designs created by designer companies are purchased by the component developer companies who supply components to the assembler companies. Each component developer company can specialise in a certain domain or set of domains.

For example, one company can specialise in building simulation components while another in data processing components. As shown in fig.1 component developer companies also exchange information with the market study companies. The decision as to which designs should be purchased will be influenced by the needs of assembler companies (identified by a market study company). This information exchange will also help component developer companies to keep up to speed with technologies being used by the assembler companies. It should also be noted that while drawing the specialisations in the example *co-operation* we considered the often-found opinion that designers should be the implementers. Nevertheless, we are of the view that the example specialisation will hold due to designer companies specialising in designing components for specific domains. Since these companies capitalise on this expertise they will produce effectively usable designs.

Since assembler companies use components developed by component developer organisations in the *co-operation* it is essential that these components be certified. This requires existence of an independent component certification body. Certification of components from such a body increases the confidence of the assembler company in the components it is purchasing. It also improves the users' confidence as the quality of the products is confirmed. It is worth noting that one component certification body can serve more than one *co-operation*.

The component assembler companies can be slightly bigger than other companies in the *co-operation*. They employ assembly lines to assemble a product from the components supplied by the component developer companies. An assembly line in such a company is similar to a *generator* in generative programming [5]. It differs in the sense that an assembly line is not fully automated. Some sensitive tasks are left to the human programmers. An assembler company can build its own assembly lines. A possible approach is to have a framework for assembly lines which can then be configured to suit building a particular product. This framework could itself be a component-based framework such as the one we proposed in [13].

One potential problem in the use of assembly lines is that they require standardised³ components especially in cases where components are being supplied by different developer companies. Software components at present do not conform to a particular standard causing compatibility and interoperability problems at the input level. However, it is an achievable task for organisations in the *co-operation* to ensure that components conform to one of the accepted industry standards e.g. COM⁴, Java Beans⁵, etc. or a standard specifically developed for the *co-operation* or those followed by the assembler organisation. Standardisation of components across the *co-operation* allows verifying the components forming the input to the assembly line. This brings in the notion of "trusted components" reducing testing costs which in turn reduces production costs and time to market. It also improves the reliability of the end software product hence increasing the customer confidence. Standardisation of components also provides effective reuse as components developed by different vendors conforming to the same specification are interchangeable. It also encourages healthy competition among suppliers further cutting production costs for the buyer. The assembly process is shown in fig. 2.

³ This reflects that specialisation induces standardisation.

⁴ <http://www.microsoft.com/com/>

⁵ <http://www.javasoft.com/beans/index.html>

As shown in fig. 2 the assembler company also specialises in sales and marketing. It chooses products to form its product lines⁶ and build its product portfolio. A product portfolio should comprise of several products at different stages of the product life cycle [11]; by carefully monitoring market developments (with the help of market study companies) the company can compete against bigger corporations by anticipating and meeting the changing market demands. The company will also make sure that there is always a better version of a product or a better new product available to replace of the older version. This will help to build customer loyalty and brand recognition of the assembler company or of the whole *co-operation*.

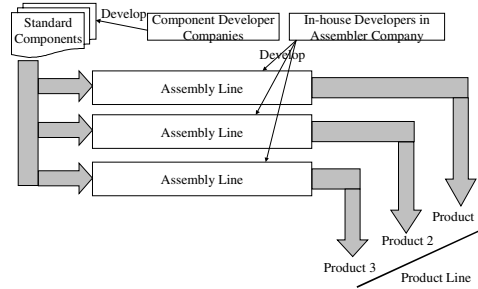


Fig. 2. The Assembly Process

3.1 The Communication Layer

As mentioned above the organisations in the *co-operation* communicate using an e-commerce infrastructure. We now describe this communication layer. As shown in fig. 3 the communication layer resides on a web server and comprises of three main processes:

- Check-in process
- Notification process
- Purchase process

We now describe some of the communications in fig. 3. The component assembler companies *check-in* their requests for market studies. All organisations in the cooperation register their interests (markets of interest for market study companies, products of interest for component assembler companies, domains of interest for designer and component developer companies, etc.) with the communication layer. Therefore, when a request for a market study is checked-in it is matched against the registered interests of market study companies and if a match is found the *notification* process notifies the appropriate market study companies. Note that it is also possible for organisations in the cooperation to query the system for information of interest e.g. in this case market study companies can query the system for market studies of interest based on keywords. There are a number of data protection, intellectual property, security and confidentiality issues raised by the communication layer. These will, however, form the subject of a future publication and will not be discussed here.

⁶ A product line is a “group of products that are closely related because they perform a similar function, are sold to the same customer group, are marketed through the same channels, or fall within given price ranges” [11].

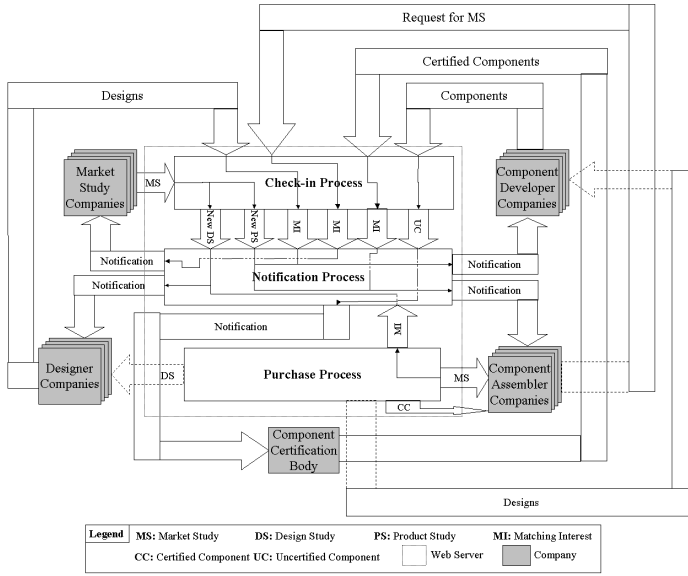


Fig. 3. E-commerce based communication layer in EProMS

Market study companies conduct market studies (as a result of a request checked-in by a component assembler company or on their own initiative) and *check* them *into* the system. For the purpose of this discussion we assume that the study checked in identifies the need for a new product. As a result the component assembler companies with matching interests are notified by the *notification* process. If the market study is of interest to a component assembler company it *purchases* the study. Once the study is purchased the *notification* process notifies any designer companies with interests in the domains addressed by the new product to be developed. These companies then may *purchase* studies identifying design requirements, develop designs and *check-in* designs which will result in appropriate *notifications* being sent to component developer companies who can then *purchase* the designs, develop components, *check-in* components and so on. Due to space limitations we will not describe any more communications. However, these can be observed from fig. 3.

4. Comparison with Existing Models

We have employed the evaluation metrics proposed by [6] in order to compare EProMS with existing models discussed earlier. The metrics are described in fig. 4 with reference to an evaluation of the waterfall model [6]. It should be noted that the function representing the users' needs in fig. 4 is neither linear nor continuous in reality. The scales on the X and Y axis have therefore not been shown and can be assumed to be non-uniform, containing areas of compression and expansion. t_0 is the time when the need for a software system is recognised and a development effort begins. t_1 is the point where an operational system is available. This system undergoes enhancements between t_1 and t_3 satisfying the original requirements at some point t_2 . At some later point in t_3 the cost of enhancement is large enough to result in a decision to build a new system. The development of the product is completed at t_4 and the cycle repeats itself. The various metrics shown in fig. 4 are described below [6]:

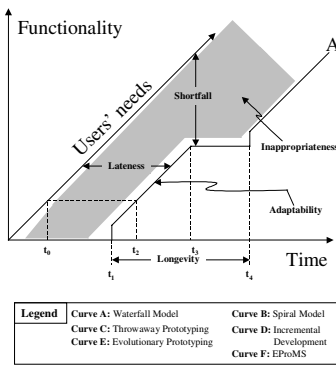


Fig. 4. Evaluation Metrics [6]

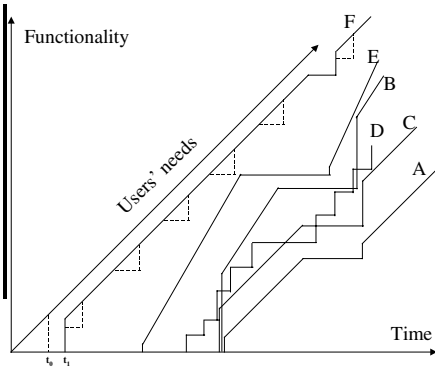


Fig. 5. EProMS in contrast with existing models

- *Shortfall* is a measure of how closely the operational system meets the actual requirements at any time t .
- *Lateness* represents the time elapsed between the emergence of a new requirement and its satisfaction.
- *Adaptability* is the rate at which the software can adapt to new requirements.
- *Longevity* is the time the product is adaptable to change after which it is replaced.
- *Inappropriateness* indicates the behaviour of shortfall over time.

These metrics were chosen in order to compare EProMS with its peers on the basis of:

- Time to market and
- Users' satisfaction

As mentioned earlier these two factors play a crucial role for the business of small organisations especially when they are competing against bigger corporations.

Fig. 5 shows our analysis of EProMS in comparison with the waterfall model, spiral model, throwaway prototyping, incremental programming and evolutionary prototyping [6]. Note that the curves have been drawn with reference to the curve for the waterfall model. In EProMS the market study companies specialise in identifying needs for new products and continuously study the users' needs in order to defend their niche by timely selling studies identifying the need for new products and components to assembler companies and designer companies respectively. As a result the initial time t_0 to identify the need for a new product is very small. Once the need for a new product is brought to the attention of an assembler company the development time for the product is quite small as the company specialises in assembly lines to build products from standard components. Also it specialises in sales and marketing. Furthermore, the initial users' requirements supplied by the market study company are identified by experienced ethnographers and technologists together and are, therefore, well captured. Due to specialist companies performing the various activities in the software development process and the elimination of a need for these companies to switch between various activities, the time to market is considerably reduced and the operational system at t_1 closely meets the user requirements. There can be a lateness introduced due to the unavailability of some components on the market. However, this need would have been identified and passed on to specialist component designer and developer companies. As a result standard components fulfilling the needs of the new product will be available on the market in a short time to be purchased by the assembler company. This lateness has been shown

by the dotted line in fig. 5. Since the market study companies will be monitoring the progress of the product after its release any changes in users' requirements will be identified promptly and catered for by the assembler company. At times a lateness can be introduced due to the unavailability of certain components but this lateness will be kept to a minimum because of specialist component designer and developer companies. As a result the curve closely follows the users' needs at all times reacting to them in a timely fashion. Due to an extensible architecture based on standardised components the product will last very long and perhaps will need to be rebuilt as a result of major technological changes in the industry. Again such changes will be identified promptly by the market study company and the lateness in building and releasing a new product will be kept to a minimum.

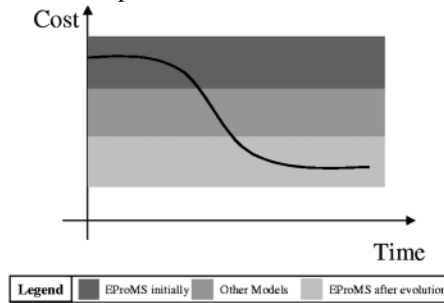


Fig. 6. EProMS: From a cost-effectiveness perspective

We have also analysed EProMS in comparison with existing models from a cost-effectiveness perspective. This analysis is shown in fig. 6. The change from existing practices to EProMS will be radical and as a result the initial cost will be high. This change will, however, need to be evolutionary as a revolutionary change will be too expensive for small organisations. Once the evolution is complete EProMS will be more cost-effective due to specialised companies concentrating their resources on a specific activity. Due to the elimination of the need to perform all (or most of) the activities resources will be used in a more efficient and effective manner. Specialisation will also allow more effective reuse of existing expert knowledge hence cutting learning costs. Also, the reduction in time to market and a high degree of users' satisfaction will increase profits for small organisations.

5. Conclusions and Future Work

We have presented a process model for software development in small organisations. We have argued that existing process models for software development are too resource-intensive to be employed by small organisations especially those competing against big corporations. Our process model is based on forming an e-commerce based *co-operation* of small organisations to produce market-driven software. Various organisations in the *co-operation* specialise in a particular aspect of software development. This allows them to capitalise on their advantages such as experience, technical knowledge, automation and reduction in delivery time due to the elimination of the need to switch between tasks. It also provides small organisations an opportunity to exploit their inherent advantages i.e. no/smaller bureaucratic apparatus, efficient communication and information exchange system, etc. Entering a *co-operation* also helps counter the recognition, credibility and stability problems faced

by small organisations. It also provides an organisation an opportunity to develop stable relations with its counterparts reducing risks such as contract completion, supply quality, timeliness, etc.

We have also provided an analysis of our model and shown that due to specialisation the *co-operation* products will follow the user needs closely and react to them in a timely fashion. There will also be a considerable reduction in the time to market. We have also discussed the cost-effectiveness of the model and shown that although the initial cost will be high it will be cost-effective once it has been fully adopted by a *co-operation*. We have discussed an e-commerce based communication layer between the various companies in the *co-operation*. Our work in the immediate future aims at making this communication layer publicly available. We aim at using it to run experiments in collaboration with small organisations in order to obtain statistical results. We will also be developing techniques for evolution from existing models to EProMS. The licensing and patenting issues raised due to the distribution of responsibilities in EProMS form another area of interest for our future research.

References

- [1] Boehm, B. W., **"A Spiral Model of Software Development and Enhancement"**, *IEEE Computer vol. 21 no. 5, August 1988*, pp. 61-72
- [2] Booch, G., *Object-Oriented Analysis and Design, 2nd Ed.*, Benjamin-Cummings, Redwood, CA, 1994
- [3] Budde, R., Kautz, K., Kuhlenkamp, K., Zuellighoven, H., *Prototyping*, Springer-Verlag, 1992
- [4] Hirsch, E., **"Evolutionary Acquisition of Command and Control Systems"**, *Program Manager, Nov.-Dec. 1985*, pp. 18-22
- [5] Czarnecki, K., Eisenecker, U. W., **"Synthesizing Objects"**, *Proceedings of ECOOP '99, LNCS 1628*, pp. 18-42, Springer-Verlag, June 1999
- [6] Davis, A. M., Bersoff, E. H., Comer, E. R., **"A Strategy for Comparing Alternative Software Development Life Cycle Models"**, *IEEE Trans. on Soft. Engg.*, Vol. 14 No. 10, Oct. 1988, pp. 1453-1461
- [7] Floyd, C., Reisin, F.-M., Schmidt, G., **"STEPS to Software Development with Users"**, *Proc. Of ESEC '89, LNCS 287*, pp. 48 - 64, Springer-Verlag 1989
- [8] Giddings, R. V., **"Accommodating Uncertainty in Software Design"**, *Communications of the ACM*, Vol. 27 No. 5, May 1984, pp. 428-434
- [9] Goma, H., Scott, D., **"Prototyping as a Tool in the Specification of User Requirements"**, *Proc. of the 5th IEEE Int. Conf. on Soft. Engg.*, 1981, pp. 333-342
- [10] Jacobson, I., Booch, G., Rumbaugh, J., **"The Unified Process"**, *IEEE Software*, May/Jun. 1999, pp. 96-102
- [11] Kotler, P., Armstrong, G., Saunders, J., Wong, V., *Principles of Marketing*, Prentice Hall Europe, 1999
- [12] McDermid, J., Ripken, K., *Life Cycle Support in the Ada Environment*, Cambridge University Press, UK, 1984
- [13] Parson, D., Rashid, A., Speck, A., Telea, A., **"A 'Framework' for Object Oriented Frameworks Design"**, *Proceedings of TOOLS '99 Europe*, pp. 141-151
- [14] Royce, W. W., **"Managing the Development of Large Software Systems: Concepts and Techniques"**, *WESTCON Technical Papers*, Vol. 14, Western Electronic Show and Convention, 1970
- [15] Viller, S., Sommerville, I., **"Coherence: An Approach to Representing Ethnographic Analysis in Systems Design"**, *Human-Computer Interaction*, Vol. 14, No. 1 & 2, 1999, pp.9-41

Extracting Object-Oriented Database Schemas from XML DTDs Using Inheritance^{*}

Tae-Sun Chung, Sangwon Park, Sang-Young Han, and Hyoung-Joo Kim

School of Computer Science and Engineering, Seoul National University
San 56-1, Shillim-dong, Gwanak-gu, Seoul 151-742, KOREA
{tschung,swpark,syhan,hjk}@oops1a.snu.ac.kr

Abstract. As XML has become an emerging standard for information exchange on the World Wide Web, it has gained attention in database communities to extract information from XML seen as a database model. Recently, many researchers have addressed the problem of storing XML data and processing XML queries using traditional database engines. Here, most of them have used relational database systems, while we show in this paper that object-oriented database systems can be another solution. Our technique generates an OODB schema from DTDs and processes XML queries. In particular, we show that the semi-structural part of XML data can be represented by ‘inheritance’ and that it can be used to improve query processing.

1 Introduction

Recently, as XML [2] has emerged as a standard for information exchange on the World Wide Web, it has gained attention in database communities to extract information from XML seen as a database model. As XML data is self-describing, we can issue queries over XML documents distributed in heterogeneous sources and get the necessary information.

There are two kinds of approaches to query XML documents. One is using special purpose query engines for semistructured data since an XML document can be regarded as an instance of a semistructured data set [3, 8, 10, 13, 15]. The other is using traditional databases such as relational databases or object-oriented databases for storing and querying XML documents [5, 7, 9, 16]. In particular, many approaches using RDBMSs have been proposed. That is, XML data is converted to relational tuples and XML queries are translated to SQL queries. However, to the best of our knowledge, there is no special work on the problem of using OODBMSs to store and query XML data. An exception is work in [5] that processes SGML data using an OODBMS to store and query SGML documents [4].

^{*} This work was supported by the Brain Korea 21 Project.

¹ Additionally, there is a special purpose XML query processor, Excelon [11] that is based on an OODBMS.

In this paper, we propose a technique that stores and queries XML data using an object-oriented database. Compared with the proposal in [5], our work differs in that we use ‘inheritance’, a key concept in object-oriented paradigms.

For example, let us assume that the following DTD is given.

```
<!ELEMENT person (name, address, vehicle*,(school|company))>
<!ELEMENT name (firstname?, lastname)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT vehicle (model, company, gear?)>
<!ELEMENT model (#PCDATA)>
<!ELEMENT gear (#PCDATA)>
<!ELEMENT school (name, baseball-team?, person+,url?)>
<!ATTLIST school name CDATA #CDATA REQUIRED>
<!ELEMENT baseball-team (#PCDATA)>
<!ELEMENT url (#PCDATA)>
<!ELEMENT company (name, person+, url?)>
<!ATTLIST company name CDATA #CDATA REQUIRED>
<!ELEMENT alumni (name, year, school)>
<!ATTLIST alumni name CDATA #CDATA REQUIRED>
<!ELEMENT year (#PCDATA)>
```

Fig. 1. An example DTD

Here, the first line says that an element *person* has the *name* and *address* sub-elements, and he or she has zero or more vehicles, and finally, is a student or a company employee. From the DTD declaration for the element *person*, we can classify the element *person* into four groups: 1. ones who have one or more vehicles and work for companies, 2. ones who have no vehicle and work for companies, 3. ones who have one or more vehicles and are students, and 4. ones who have no vehicle and are students. Our technique uses this information in designing object-oriented schema by means of inheritance semantics. In the above example, each group is defined as *Person-1*, *Person-2*, *Person-3*, and *Person-4* type classes that inherit the general class *Person*. Here, for example, as *Person-1* is a specialization of *Person*, the inheritance semantics is satisfied.

If we design object-oriented schemas in this way, it can be used for enhancing query evaluation. For example, if a query is related to students having vehicles, a query processor can only traverse extents of *Person-3*.

This paper shows a technique of extracting OODB schemas using inheritance and querying XML documents stored in an OODBMS.

2 Deriving an OO Schema from a DTD

In previous work of [5], each class is created for each element definition. Here, the choice operator(‘|’) is modeled by a union type, and the occurrence indicators

(‘+’ or ‘*’) are represented by lists. Values (e.g. strings) of XML data are represented by O_2 classes of appropriate content types (e.g., Text) using inheritance. Figure 2 shows an object-oriented schema for the DTD in Figure 1.

```
class Person public type tuple(name:Name, address:Address,
                             vehicle:list(Vehicle),union(school:School,company:Company))
class Name public type tuple(firstname:Firstname,lastname:Lastname)
class Firstname inherit Text
class Lastname inherit Text
...
```

Fig. 2. An OODB schema

However, the technique has several problems as follows.

- Since each element definition creates one class, several classes are created though the classes can be inlined into one class. For example, in Figure 2, the classes *Name*, *Firstname*, and *Lastname* can be inlined into the class *Person*.
- The technique does not use inheritance in designing classes. For example, if we design the *Person* class in the real world, we create a class *Person* as a base class and create classes *Student* and *Employee* that are subclasses of the class *Person*. In this case, query processing can be improved. For example, if a query is only targeted to the class *Student*, a query processor can only traverse objects of the class *Student*. It need not traverse all of the *Person* type classes.
- For the occurrence indicator (‘*’) or optional indicator (‘?’), if an object has no value at the corresponding field, the field should be set to null. This has drawbacks in memory efficiency.
- In the technique, the choice operator (‘|’) is modeled by a union type. However, since the ODMG [4] model, which is a standard for object database management systems, does not support union of types, it can not be applied to ODMG-compliant object-oriented databases directly.

So, our approach solves the above problems in the following ways:

- By applying an inlining technique of relational databases, we inline as many descendants of an element as possible into a single class(Section 2.1).
- After classifying DTD elements, we reconstruct classes using inheritance(Section 2.2).

2.1 Class Inlining Technique

We adopt the inlining technique of relational databases proposed in [16]. The technique in [16] inlines as many descendants of an element as possible to a single relation.

Compared with the work in [16], our technique has two different points. First, as the traditional relational databases do not support set-valued attributes, when

an element has a sub-element with ‘+’ or ‘*’ expression, the sub-element is made into a separate relation and the relationship between the element and the sub-element is represented by introducing a foreign key. For example, in Figure 3, a relation for “vehicle” is created and links from vehicles to persons are created by using foreign keys. In an object-oriented model, as an occurrence indicator(‘*’ or ‘+’) can be represented by lists, we don’t have to introduce a foreign key manually. In the above example, the class *Person* can be represented by having a set-valued attribute *vehicle*.

Second, in relational models, to represent relationships between relations, join attributes should be created manually. However, in object-oriented models, as relationships between classes can be represented by direct pointers, manual join attributes don’t have to be created. For instance, in Figure 3, when the relations for “alumni” and “school” are created in relational models, the relation *School* has a foreign key *parentId* that joins schools with alumni. In object-oriented models, the class *Alumni* has an attribute *school* that has object identities of the class *School*.

When given a DTD, the class inlining technique creates an object-oriented schema as follows. First, Figure 3 shows a DTD graph for the DTD in Figure 1. A DTD graph introduced in [16], represents the structure of a DTD.

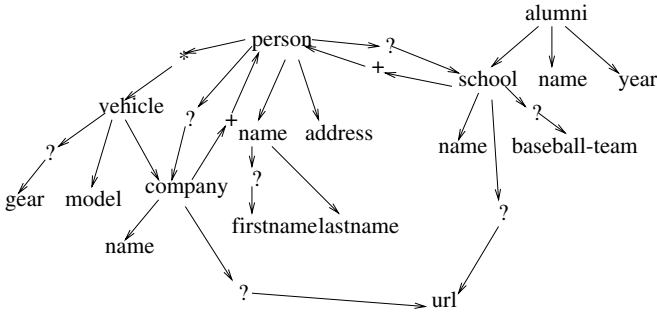


Fig. 3. A DTD graph

Next, we decide what classes to create from the DTD graph by the following rules.

1. Classes are created for element nodes that have an in-degree of zero. Otherwise, the element can not be reached. For example, the class for “alumni” is created, because the element node *alumni* has an in-degree of zero.
2. Elements below a ‘*’ or a ‘+’ node are made into separate classes. This is necessary for classes that have set-valued attributes. For instance, the element node *vehicle* that is below a ‘*’ node is made into a separate relation. The class *Person* will have a set-valued attribute *vehicle*.
3. Nodes with an in-degree of one are inlined. For example, in Figure 3, nodes *gear* or *model* are inlined as they have an in-degree of one.
4. Among mutually recursive elements all having in-degree one, one of them is made into a separate relation.
5. Element nodes having an in-degree greater than one are made into separate relations.

If we apply these five rules to the DTD graph in Figure 3, the classes *Person*, *Vehicle*, *Company*, *School*, *Alumni*, and *Url* are created. Once we decide which classes are created, we construct an object-oriented schema. In the DTD graph, if X is an element node that is made into a separate class, it inlines all the nodes Y that are reachable from it such that there is no node that is made into a separate class in the path from X to Y . An object-oriented schema is created for the DTD graph in Figure 3 as follows.

```
class Person public type tuple(name.firstname:string,name.lastname:string,
    address:string,vehicle:list(Vehicle),school:School,company:Company)
class School public type tuple(name:string,baseball-team:string,
    person:list(Person),url:Url)
class Alumni public type tuple(name:string, year:String,school:School)
class Company public type tuple(name:string,person:list(Person),url:Url)
class Url inherit Text
class Vehicle public type tuple(model:string,company:Company,gear:string)
```

Fig. 4. An OODB schema

2.2 Designing a Schema Using Inheritance

XML data having irregular schema can be represented by inheritance, because if we design the structural part of an element as a superclass and the semi-structural part of it as subclasses, the generalization relationship between the superclass and the subclasses is satisfied.

DTD Automata. First, for each element that is made into a separate class, we abstract a DTD as a set of $(n : P)$ pairs. Here, let N be a set of element names, $n \in N$, and P is either a regular expression over N or PCDATA which denotes a character string.

For an element e and the corresponding DTD declaration $(n : P)$, the regular expression P can be divided into five categories as follows. If r , r_1 , and r_2 are regular expressions that DTDs represent, $L(r)$, $L(r_1)$, and $L(r_2)$ are the languages that can be described by the regular expressions.

1. case $r = r_1, r_2$: The languages that r denotes are the concatenation of $L(r_1)$ and $L(r_2)$.
2. case $r = r_1|r_2$: $L(r)$ is the union of $L(r_1)$ and $L(r_2)$.
3. case $r = r_1^+$: This represents more than one repetition of the same structure.
4. case $r = r_1^*$: This is the same as case 3 except that it permits zero repetitions of the same structure.
5. case $r = r_1^?$: This represents zero or one occurrence of the same structure.

Among these five categories, cases 2,4, and 5 result in subclasses, while case 1 and 3 don't result in subclasses. This is because in cases 1 and 3, XML data that conforms to the DTD has attributes of the same form. The reason case 4 becomes information is because whether or not an element has an attribute can be represented by specialization. On the other hand, whether an element has one attribute or more than one can not be explained by specialization.

So, we define the following relaxed regular expression to extract only the necessary information in classifying elements.

Definition 1 (Relaxed Regular Expression). *A relaxed regular expression is constructed from a given regular expression as follows.*

1. $r_1, r_2 \Rightarrow r_1, r_2$
2. $r_1 | r_2 \Rightarrow r_1 | r_2$
3. $r+ \Rightarrow r$
4. $r* \Rightarrow r + | \perp \Rightarrow r | \perp$ (by rule 3)
5. $r? \Rightarrow r | \perp$

Example 1. In Figure 1 the DTD declaration for the element *person* is abstracted to (person: (name, address, vehicle*, (school|company))), and we get (person : (name, address, (vehicle| \perp), (school|company))) after applying the relaxed regular expression.

DTD automata are constructed in the following ways. Let $(n_i : P'_i)$ be an expression which is obtained by applying relaxed regular expressions to each DTD declaration $(n_i : P_i)$. We construct automation A_i by Algorithm 1 with a new regular expression $n_i P'_i$.

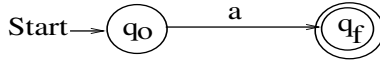


Fig. 5. $r = a$

Theorem 1. *There always exists an automaton M constructed by Algorithm 1 for the input regular expression r , and if $L(M)$ is the language accepted by M , and $L(r)$ is the language which is describable by the regular expression r , then $L(M) = L(r)$.*

We omit the proof for lack of space.

Example 2. Figure 6 shows an automaton constructed by Algorithm 1 for the element *person* after applying the relaxed regular expression, i.e. (person : (name, address, (vehicle| \perp), (school|company))).

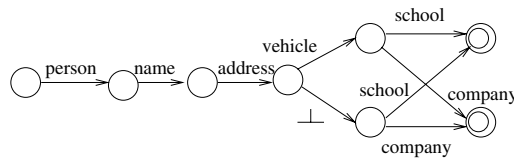
² In this paper, we occasionally omit the concatenation operator, that is, $n_i P'_i = n_i, P'_i$.

Algorithm 1 The construction of DTD automata

```

1: Input: A relaxed regular expression  $r$ 
2: Output: An automaton  $M$ 
3: procedure Make_DTD_Automata(regular expression  $r$ )
4: if  $r = a$  ( $a \in \Sigma$ ) then
5:   Construct an automaton  $M$  as shown in Figure 5
6:   return  $M$ ;
7: else if  $r = r_1|r_2$  then
8:    $M_1 = (Q_1, \sum_1, \delta_1, q_1, F_1) \leftarrow \text{Make\_DTD\_Automata}(r_1)$ ;
9:    $M_2 = (Q_2, \sum_2, \delta_2, q_2, F_2) \leftarrow \text{Make\_DTD\_Automata}(r_2)$ ;
10:  Construct the new automaton  $M = (Q_1 - \{q_1\} \cup Q_2 - \{q_2\}, \sum_1 \cup \sum_2, \delta, [q_1, q_2], F_1 \cup F_2)$  from the automata  $M_1$  and  $M_2$ , where  $\delta$  is defined by
      1.  $\delta(q, a) = \delta_1(q, a)$  for  $q \in Q_1 - \{q_1\}$  and  $a \in \sum_1$ ,
      2.  $\delta(q, a) = \delta_2(q, a)$  for  $q \in Q_2 - \{q_2\}$  and  $a \in \sum_2$ ,
      3.  $\delta([q_1, q_2], a) = \delta_1(q_1, a)$  where  $a \in \sum_1$ ,
      4.  $\delta([q_1, q_2], a) = \delta_2(q_2, a)$  where  $a \in \sum_2$ ;
11: else  $\{ r = r_1, r_2 \}$ 
12:    $M_1 = (Q_1, \sum_1, \delta_1, q_1, F_1) \leftarrow \text{Make\_DTD\_Automata}(r_1)$ ;
13:    $M_2 = (Q_2, \sum_2, \delta_2, q_2, F_2) \leftarrow \text{Make\_DTD\_Automata}(r_2)$ ;
14:   Let the final states  $F_1$  of  $M_1$  be states  $f_1, f_2, \dots, f_m$  ( $m \geq 1$ ).
      Construct the new automaton  $M = (Q_1 - F_1 \cup Q_2 - \{q_2\} \cup \{[f_1, q_2], [f_2, q_2], \dots, [f_m, q_2]\}, \sum_1 \cup \sum_2, \delta, q_1, F_2)$  from the automata  $M_1$  and  $M_2$ ,
      where  $\delta$  is defined by
      1.  $\delta(q, a) = \delta_1(q, a)$  for  $q \in Q_1 - F_1$ ,  $\delta_1(q, a) \neq f_k$  (where  $1 \leq k \leq m$ ), and  $a \in \sum_1$ ,
      2.  $\delta(q, a) = \delta_2(q, a)$  for  $q \in Q_2 - q_2$  and  $a \in \sum_2$ ,
      3.  $\delta([f_k, q_2], a) = \delta_2(q_2, a)$  for all  $k$  (where  $k = 1, 2, \dots, m$ ) and  $a \in \sum_2$ ,
      4.  $\delta(q_f, a) = [f_k, q_2]$  for all  $q_f$  which satisfies  $\delta_1(q_f, a) = f_k$  (where  $1 \leq k \leq m$ ) and  $a \in \sum_1$ ;
15: end if
16: return  $M$ 

```

**Fig. 6.** A DTD automaton

Classification of DTD elements using DTD automata. As the DTD automata are constructed from relaxed regular expressions, they contain information only about concatenations and unions. Here, the diverging points in automata become those of classifying DTD elements. So, by recording the labels at diverging points we can classify the DTD elements.

Figure 7 shows a classification tree from the DTD automaton of the element *person* in Figure 6 and the corresponding classification table. Here, the DTD element *person* is divided into 4 groups according to its label sets, namely {*vehicle*, *school*}, {*vehicle*, *company*}, {*school*}, and {*company*}.

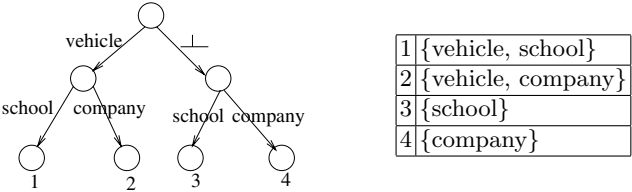


Fig. 7. A classification tree and a classification table

The fact that each element is classified into several groups stems from the flexibility of XML data, and it becomes a hint to a query processor. This is because if a query is targeted to certain groups only, a query processor can only traverse the groups targeted.

We created a superclass from the structural part of an element and several subclasses that inherit the class. The subclasses are created from the semi-structural part of the element. For example, for the element *person*, the class *Person* has attributes *name* and *address*. Further, subclasses are created that inherit it and has attributes {*vehicle*, *school*}, {*vehicle*, *company*}, {*school*}, and {*company*}. In this way, an object-oriented schema is created as follows.

```
class Person public type tuple(name.firstname:string,name.lastname:string,
    address:string)
class Person1 inherit Person type tuple(vehicle:list(Vehicle),
    school:School)
class Person2 inherit Person type tuple(vehicle:list(Vehicle),
    company:Company)
class Person3 inherit Person type tuple(school:School)
class Person4 inherit Person type tuple(company:Company)
...
```

Fig. 8. An OODB schema

3 Query Language

Several query languages including XML-QL [6], UnQL [3], Lorel [1], and XQL (from Microsoft) have been proposed to query semistructured data. The semistructured query languages are based on the following regular path expression that expands the path expression [12] of object-oriented database query languages.

Definition 2. (*Regular Path Expression*) A regular path expression is in the form of $H.P$, where

1. H is an object name or a variable denoting an object,
2. P is a regular expression over labels in an DTD, i.e. $P = \text{label} | (P|P) | (P.P) | P^*$.

We should solve two problems to convert semistructured query languages based on regular path expressions to object-oriented query languages. First, as object-oriented query languages do not support processing the alternation operator ($|$) that stems from $(P|P)$ in the regular path expression, we should make a routine that processes it. We adapt the alternation elimination technique proposed in [14]. For example, the regular path expression $\text{person}.\text{(school|company)}.\text{name}$ is converted to $(\text{person.school.name}) \cup (\text{person.company.name})$.

Second, we should process arbitrary complex queries, i.e. nested recursive queries that stems from (P^*) in the regular path expression. In [14], the authors suggest a technique that replaces all regular expression operators with possible path instantiations using DataGuides [10] that are structural summaries of databases. We suggest a similar technique using DTDs. First, we define a simple regular path expression as follows.

Definition 3. (*Simple Regular Path Expression*) A simple regular path expression is a sequence $H.p_1.p_2....p_n$ where

1. H is an object name or a variable denoting an object,
2. p_i (where $1 \leq i \leq n$) is a label in an DTD or wild-card $*$ which denotes any sequence of labels.

Compared to the regular path expression, it is simple, but can process almost all XML queries. We show that queries that have simple regular path expressions can be converted to object-oriented queries. Our technique can be generalized to regular path expressions.

3.1 Translating Simple Regular Path Expressions without the $*$ Operator

In this section, we deal with simple regular path expressions without the $*$ operator. In this case, we can convert it to object-oriented database query languages easily. For example, consider the following Lorel-like semistructured query.

```
select X.name.firstname, X.name.lastname
from person X, X.vehicle Y
where X.address = "Seoul", Y.model = "EF-Sonata", Y.gear="auto"
```

The query asks for the first and last name of the person who has a vehicle “EF-Sonata” with an automatic transmission. The query is converted to the following object-oriented database query languages.

```
select tuple(f:p."name.firstname",l:p."name.lastname")
from p in Person,y in p.Vehicle
where p.address = "Seoul", y.model = "EF-Sonata", y.gear="auto"
```

When the query is processed, the number of target classes can be reduced. That is, in the database schema in Figure 8, as the variable p bound to the class *Person* has an attribute *vehicle*, only the instances of the class *Person1*, *Person2* are traversed.

3.2 Converting Simple Regular Path Expressions with the ‘*’ Operator

Queries having the expression ‘*’ that denotes any sequence of paths are frequently used in XML queries by those who do not know database schema. For example, consider the following query.

```
select u
from person.*.url u
```

The query requires all urls that are reachable by the paths that first, have the edge *person* followed by any sequence of arbitrary edges, and next, have the edge *url*. As object-oriented database query languages do not support this kind of queries directly, we convert the path expression with the ‘*’ operator to possible path instantiations using the DTD graph in Section 2.1. Here, operators in the DTD graph are excluded.

The ‘*’ expression in the above query is replaced with the paths *school*, *company*, and *vehicle.company*. Thus, the query is converted to the following Lorel-like query.

```
select u
from (person.school.url|person.company.url|
      person.vehicle.company.url) u
```

Next, The query is converted to the following object-oriented query.

```
select u
from p in Person,s in p.school,c in p.company,v in p.vehicle
      v2 in v.company, u in (s.url,c.url,v2.url)
```

4 Conclusion

In this paper, we showed that object-oriented databases can be another solution for storing and querying XML data. We propose a technique that creates object-oriented schemas from DTDs. In particular, we solve the problem of impedance

mismatch that stems from the flexibility of XML data by using inheritance. That is, by representing the semi-structural part of XML data using inheritance, our technique solves the null value problem and enhances query processing.

We showed that XML queries composed of simple regular path expressions are converted to object-oriented database query languages and the results of queries to XML data. Here, we suggest a technique that removes the ‘*’ expression by using DTDs.

References

1. S. Abiteboul, Dallan Quass, Jason McHugh, Jennifer Widom, and Janet Wiener. The lorel query language for semistructured data. *International Journal on Digital Libraries*, 1996.
2. T. Bray, J. Paoli, and C. Sperberg-McQueen. Extensible markup language (XML) 1.0. Technical report, W3C Recommendation, 1998.
3. Peter Buneman, Susan Davidson, Gerd Hillebrand, and Dan Suciu. A query language and optimization techniques for unstructured data. In *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, 1996.
4. R.G.G. Cattell. *The object database standard: ODMG-93*. Morgan Kaufmann Publishers, 1994.
5. V. Christophides, S. Abiteboul, S. Cluet, and M. Scholl. From Structured Documents to Novel Query Facilities. In *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, 1994.
6. A. Deutsch, M. Fernandez, D. Florescu, A. Levy, and D. Suciu. Query language for XML. In *Proceedings of Eighth International World Wide Web Conference*, 1999.
7. Alin Deutsch, Mari Fernandez, and Dan Suciu. Storing semistructured data with STORED. In *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, 1999.
8. Mary Fernandez and Dan Suciu. Optimizing regular path expressions using graph schemas. In *IEEE International Conference on Data Engineering*, 1998.
9. Daniela Florescu and Donald Kossmann. Storing and querying XML data using an RDBMS. *IEEE Data Engineering Bulletin*, 1999.
10. Roy Goldman and Jennifer Widom. DataGuides: enabling query formulation and optimization in semistructured databases. In *Proceedings of the Conference on Very Large Data Bases*, 1997.
11. <http://www.odi.com/excelon>. 2000.
12. M. Kifer, W. Kim, and Y. Sagiv. Querying object-oriented databases. In *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, 1992.
13. J. McHugh, S. Abiteboul, R. Goldman, D. Quass, and J. Widom. Lore: A Database management system for semistructured data. *SIGMOD Record*, 1997.
14. J. McHugh and J. Widom. Compile-Time Path Expansion in Lore. In *Proceedings the Workshop on Query Processing for Semistructured Data and Non-Standard Data Formats*, 1999.
15. Tova Milo and Dan Suciu. Index structures for path expressions. In *Proceedings of the International Conference on Database Theory*, 1999.
16. Jayavel Shanmugasundaram, H. Gang, Kristin Tufte, Chun Zhang, David DeWitt, and Jeffrey F. Naughton. Relational Databases for Querying XML Documents: Limitations and Opportunities. In *Proceedings of the Conference on Very Large Data Bases*, 1999.

Creating XML Documents from Relational Data Sources

César M. Vittori, Carina F. Dorneles, and Carlos A. Heuser

Universidade Federal do Rio Grande do Sul - UFRGS
Instituto de Informática
Caixa Postal 15064 - CEP 91501-970
Porto Alegre - RS - Brazil
{cvittori,dorneles,heuser}@inf.ufrgs.br

Abstract. In various application, it may be necessary to obtain XML representation from data maintained in relational databases. This paper presents a language called XML/SQL which is aimed at the specification of XML documents extracted from the results of relational queries. XML/SQL allows the creation of documents of arbitrary structure regardless of the structure of the relational data source. Further, XML/SQL is a declarative language, i.e., no procedural specification of the result is required.

1 Introduction

XML [BPSM98] is becoming a standard for the representation and exchange of data through the WEB. A large proportion of existing data are still stored in traditional databases. In many cases, it is necessary to transform data stored in traditional databases into the XML format. That happens, for example, when one wishes to generate WEB pages from the contents of a database, when it is necessary to exchange data stored in a database with other applications, or when applications process data internally using XML format.

This paper tackles the problem of transforming data stored in a relational databases into documents in XML format.

One approach to this problem is to use a XML query language, such as XML-QL [D+98], XQL [T+98], or Quilt [CRF00]. In that approach, the relational database model must be mapped into XML. The queries submitted in a XML query language must be mapped into SQL. One example of such approach is SilkRoute [FTS00]. This type of solution is suitable for users who are familiar with the XML model and its query language.

Another approach consists of using SQL queries and mapping the resulting relational views into the XML model. In that approach, the user knows the schema of the relational database. Such approach is suitable for users who are familiar with the relational model and the SQL language. This is the approach followed here, as it is directed to programmers of applications that access relational databases.

Several tools that follow this approach have been proposed [Bou00]. In many of them, the structure of the resulting XML document is restricted by the structure of the relational database view. The structure of the resulting XML documents resembles the flat structure of relational views. No nesting of collections is allowed. Examples of tools that fit in this category are DB2XML [Vol99], Oracle XML SQL Utility [ORA99,Wai99], as well as extension of the APIs SAX and DOM for databases [Lad01].

Tools that allow the specification of resulting documents with nesting of collections are the ODBC2XML tool [ISR00] and the IBM DB2 XML Extender tool [CX00,IBM00].

The limitation of ODBC2XML is that the specification of the result is procedural and, for complex documents, involves the execution of interactive SQL queries. This can compromise the efficiency of the execution of the query, because the query optimizer of the relational DBMS is not used.

In the IBM DB2 XML Extender tool, each query consists of a single SELECT SQL command. This limits the expressive power of the language when dealing with several 1:n relationships involving the same entity (see example in Section 2.3).

In this paper, we describe XML/SQL, a language for creating XML documents from relational data sources. XML/SQL allows the specification of arbitrary XML documents with several levels of nesting, by means of a flexible mapping from relations to XML documents. The query itself as well as its result are XML documents.

This way, the XML/SQL language provides a flexible mechanism for querying and extracting data from relational sources. By the use of an XML/SQL interpreter, a semi-automatic mechanism for the generation of relational to XML wrappers [PGMW95] is provided.

This paper is structured as follows. Section 2 presents the XML/SQL language through examples given in order of increasing complexity. Initially (in Section 2.1), we give an example query without nesting which reflects the flat structure of the relational view. Afterwards, we present queries which involve nesting of collections (Section 2.2) and nesting of several collections in the same element (Section 2.3). Section 3 contains the concluding remarks. The DTD of the XML/SQL language is given in Appendix A.

2 XML/SQL

We use the database shown in Figure 1 in the examples of XML/SQL queries that follow. That database comprises departments, professors, and courses. Each department is described by a department ID, a name, and an acronym. The acronym for department with id 2 is not given (NULL value). For each department, there are several courses associated with it, and each course is described by a course ID, a name, and the department ID to which it belongs. There exist several professors in each department, and professors are described by a profes-

Professor			Department		
IdProf	ProfName	IdDept	IdDept	DeptName	Acr
1	Ann	1	1	Computer Science	CS
2	John	1	2	Philosophy	NULL
3	Bill	2			

Course			ProfCourse	
IdCourse	IdDept	CourseName	IdProf	IdCourse
1	1	Database Systems	1	1
2	2	Philosophy I	1	3
3	1	Compilers	2	1
			3	2

Fig. 1. Database used in examples

sor ID, a name, an the department ID for which they work. Also, a course is taught by several professors, and one professor teaches several courses.

2.1 Creating Documents with Flat Structure

An XML/SQL query is a valid XML document, according to the XML/SQL DTD presented in Appendix A (see structure in Figure 2).

```
<XMLSQL>
  <QUERY>
    ...
  </QUERY>
  <CONSTRUCT>
    ...
  </CONSTRUCT>
</XMLSQL>
```

Fig. 2. Structure of an XML/SQL query

- The root element is the XMLSQL element. This element contains two elements:
- the QUERY element, which encloses the SQL queries to the relational database, and
 - the CONSTRUCT element, which contains the specification of the XML document to be created from the SQL queries.

The QUERY clause consists of a set of SQL instructions, which are executed over *base relations* generating *base views* as result. Each SQL instruction is enclosed within an SQL element.

The CONSTRUCT clause specifies the XML document resulting from the base views.

Using the database shown in Figure 1, the XML/SQL query below produces a list of departments.

```

<XMLSQL version="1.0">
  <QUERY>
    <SQL idsql="v1">SELECT d.iddept, d.deptname, d.acr
                        FROM department d
    </SQL>
  </QUERY>
  <CONSTRUCT>
    <LIST tagname="Departments" idsql="v1">
      <SEQUENCE tagname="Department">
        <ATOM tagname="Dept" source="iddept"/>
        <ATOM tagname="Name" source="deptname"/>
        <ATOM tagname="Acronym" source="acr" mandatory="yes"/>
      </SEQUENCE>
    </LIST>
  </CONSTRUCT>
</XMLSQL>

```

In this query, the SQL instruction generates a base view whose schema is `v1(iddept,deptname,acr)`. This relation holds the ID, the name, and the acronym of each department. The `idsql` attribute of the SQL element identifies that base view.

The XML/SQL result produced by the previous query execution is as follows:

```

<Departments>
  <Department>
    <Dept>1</Dept>
    <Name>Computer Science</Name>
    <Acronym>CS</Acronym>
  </Department>
  <Department>
    <Dept>2</Dept>
    <Name>Philosophy</Name>
    <Acronym></Acronym>
  </Department>
</Departments>

```

The result is a list of departments. Each `Department` element corresponds to a department tuple in the base view. A `Department` element is composed of a sequence of elements which correspond to the base view attributes.

In the specification of the result of a XML/SQL query, the following *constructors* can be used:

- **LIST**: defines a list (collection) of elements of the same type;
- **SEQUENCE**: defines a sequence of elements, possibly of varying types;
- **ATOM**: defines a *text-only* element.

In the example query, the `tagname` attribute from the **LIST** constructor defines the name `Departments` for the element generated as root in the XML/SQL result. Similarly for the **SEQUENCE** and **ATOM** constructors, the `tagname` attribute

gives names to elements. This attribute is required in all constructors. The `idsql` attribute from the `LIST` constructor makes reference to the base view which contains the corresponding data. In the particular case of that example, `idsql="v1"` makes reference to the base view which contains the department tuples. The `LIST` constructor generates a component element for each tuple in the base view.

In that same case, the `SEQUENCE` constructor specifies the element that will be generated for each tuple in the base view. In that example, each `SEQUENCE` corresponds to an element composed of three elements which are specified by the `ATOM` constructor.

The `source` attribute of the `ATOM` constructor makes reference to the attribute in the corresponding base view. In the example, the value for the `iddept` attribute in one line of the base view `v1` is used as content for a `Dept` element generated in the XML/SQL result.

The `mandatory` attribute in the `ATOM` constructor is used for handling null values. When the `mandatory` attribute is assigned `yes`, an element will always be generated in the result, regardless of the value of the attribute in the base view being `NULL` or not. When the `mandatory` attribute is assigned `no` (which is the “default” value), an element will only be generated in the query result in case the attribute value in the base view is different from `NULL`. For example, in the case of the `acr` attribute of the base view `v1`, the `ATOM` constructor’s `mandatory` attribute assigned value `yes` forces the generation of an `Acronym` element (see department 2 in the example).

A valid result specification is formed by a combination of the constructors defined above according to the XML/SQL DTD, given in the Appendix.

Following the XML formation rules, the result of a XML/SQL query execution has a single root element. As a consequence, the result specification always contains a single `LIST` constructor, which corresponds to the root of the XML/SQL result. This way, the XML/SQL result will always be a list.

A `LIST` constructor has a single child constructor which defines the type of the elements in the list. Three types of lists can be defined according to the child constructor:

- `LIST`: list of lists
- `SEQUENCE`: list of complex objects (a complex object is defined as a sequence of constructors)
- `ATOM`: list of values (textual elements)

A `SEQUENCE` constructor has an arbitrary sequence of child constructors. An `ATOM` constructor has no child constructors.

2.2 Nesting Elements

In the previous example, the XML document reflects the planar structure of a relation. The root element is a list of sequences of atoms.

The XML/SQL language allows the creation of XML documents with arbitrary nesting of elements. As an example, consider the query below:

```

<QUERY>
  <SQL idsql="v2">SELECT d.iddept, d.deptname, c.idcourse, c.coursename
                    FROM department d, course c
                    WHERE d.iddept = c.iddept
                    ORDER BY d.iddept
  </SQL>
</QUERY>

```

This queries retrieves, for each course, the department identifier, the department name, the course identifier and the course name. The result is sorted by department identifier.

Consider that we wish to write an XML/SQL query which, for the database in Figure 1, obtains the following result:

```

<Departments>
  <Department>
    <Dept>1</Dept>
    <Name>Computer Science</Name>
    <Courses>
      <Course>Database Systems</Course>
      <Course>Compilers</Course>
    </Courses>
  </Department>
  <Department>
    <Dept>2</Dept>
    <Name>Philosophy</Name>
    <Courses>
      <Course>Philosophy I</Course>
    </Courses>
  </Department>
</Departments>

```

In this document, for each department, the courses from that department are shown.

In order to get that result on the base view v2 above, we need the CONSTRUCTOR clause below:

```

<CONSTRUCTOR>
  <LIST tagname="Departments" idsql="v2" nestby="iddept">
    <SEQUENCE tagname="Department">
      <ATOM tagname="Dept" source="iddept"/>
      <ATOM tagname="Name" source="deptname"/>
      <LIST tagname="Courses" idsql="v2">
        <ATOM tagname="Course" source="coursename"/>
      </LIST>
    </SEQUENCE>
  </LIST>
</CONSTRUCTOR>

```

In the example above, the **nestby** attribute of the **LIST Departments** constructor is the attribute's name (**iddept**) in the base view **v2** which identifies each **Department** object. For each value of this attribute in the base view, an element from the list will be generated; in this case, a **Department** element. That is, it is possible that an XML element will be generated from various tuples from the base view.

Note that in this case the use of the **ORDER BY** clause in the SQL query is required, so as to allow the identification of the tuples which compose an XML element, when it runs through the base view. The **nestby** attribute consists of a list of attribute names from the base view.

The generated **Department** element is a sequence comprising:

1. an atomic element **Dept** taken from the **iddept** attribute
2. an atomic element **Name** taken from the **deptname** attribute, as well as,
3. a list element named **Courses**.

This list is composed of atomic elements, one for each course from the respective department. As the specification of this list does not have the **nestby** attribute, an element from the list is generated for each tuple from the base view.

There is no limitation regarding the number of lists in each nesting level, nor regarding the number of levels of nesting (except the restrictions imposed on particular implementations).

2.3 Several Lists at the Same Level

As an example of an XML document in which an element has two lists, consider the document below, which is to be obtained from the database in Figure [11](#).

```
<Departments>
  <Department>
    <Name>Computer Science</Name>
    <DeptProfessors>
      <Professor>Ann</Professor>
      <Professor>John</Professor>
    </DeptProfessors>
    <Courses>
      <Course>
        <Name>Database Systems</Name>
        <CourseProfessors>
          <Professor>Ann</Professor>
          <Professor>John</Professor>
        </CourseProfessors>
      </Course>
      <Course>
        <Name>Compilers</Name>
        <CourseProfessors>
          <Professor>Ann</Professor>
```

```

        </CourseProfessors>
    </Course>
</Courses>
</Department>
<Department>
    <Name>Philosophy</Name>
    ...
</Departments>

```

As this query involves two different traversals of relationships (from each department to its lecturers, and from each department to its courses), it is not efficient to use a single SQL query. The base view that results from such a query is not in the fourth normal form, i.e. the result contains a cartesian product of the department lecturers with the department courses.

The solution chosen in XML/SQL is the use of multiple base views, i.e., the use of multiple SQL queries, one for each traversal of a relationship.

In the example, two base views are necessary, given that there are two lists (**DeptProfessors** and **Courses**) composing the same XML element (**Department**). The **QUERY** clause in that case is as follows:

```

<QUERY>
  <SQL idsql="v3">SELECT d.iddept, d.deptname, p.idprof, p.profname
                    FROM professor p, department d
                    WHERE d.iddept = p.iddept
                    ORDER BY d.iddept
  </SQL>
  <SQL idsql="v4">SELECT d.iddept, c.idcourse, c.coursename,
                    p.idprof, p.profname
                    FROM department d, course c, profcourse pc,
                    professor p
                    WHERE d.iddept = c.iddept AND
                    c.idcourse = pc.idcourse AND
                    pc.idprof = p.idprof
                    ORDER BY d.iddept, c.idcourse
  </SQL>
</QUERY>

```

One base view (**v3**) contains a tuple for each lecturer, together with the information about the department. The same is sorted by department identifier. The other base view (**v4**) contains a tuple for each lecturer of a course, alongside the course data and the department data. This base view is sorted by department identifier and by course identifier. Two sorting attributes are needed, since there are two nesting levels (**Departments** have **Courses** which, in their turn, have **CourseProfessors**).

We give below the **CONSTRUCTOR** clause for the referred query:

```

<CONSTRUCTOR>
  <LIST tagname="Departments" idsql="v3" nestby="iddept">
    <SEQUENCE tagname="Department">

```

```

<ATOM tagname="Name" source="deptname"/>
<LIST tagname="DeptProfessors" idsql="v3">
  <ATOM tagname="Professor" source="profname"/>
</LIST>
<LIST tagname="Courses" idsql="v4" nestby="idcourse">
  <SEQUENCE tagname="Course">
    <ATOM tagname="Name" source="coursename"/>
    <LIST tagname="CourseProfessors" idsql="v4">
      <ATOM tagname="Professor" source="profname"/>
    </LIST>
  </SEQUENCE>
</LIST>
</SEQUENCE>
</LIST>
</CONSTRUCTOR>

```

When an XML/SQL query is built on more than one base view, it is necessary to specify a criterion with which the results from the two queries will be combined in forming the XML elements. This role is fulfilled by the `nestby` attribute, which not only serves as an attribute for grouping tuples from a base view, but it also serves as criterion for matching lines from one base view with lines from the other.

In the example, an XML element `Department` is formed by elements from both base views (`v3` and `v4`). In that case, the relational attribute `iddept` (specified by the `nestby` attribute) serves as criterion for:

- grouping lines from the base view `v3`,
- grouping lines from the base view `v4`, and
- matching every group formed from base view `v3` with the groups formed from base view `v4`.

In the remaining aspects, the query behaves as the previous examples. Differently from those examples, though, there are now two nesting levels, indicated by the nested `LIST` constructor.

3 Concluding Remarks and Future Work

The XML/SQL was conceived as a query language for an access layer to legacy relational databases. This layer was intended for building complex objects needed in a transaction, based on declarative specifications. The motivation is to remove from programs the burden of creating complex objects, normally specified in a procedural fashion, which involve several SQL queries and handling of cursors over those queries.

Therefore, the XML/SQL language was designed taking into considerations the following aspects:

- *Availability of a higher abstraction level*

A software module can use the XML/SQL language to access a database

and create complex XML objects. The structure of the built objects is not determined by the structure of the database; it is specified in a declarative fashion.

- *Efficiency in the access to relational sources*

XML/SQL uses SQL to build and extract data from relational sources. Thus, it can profit from the query optimisation available in relational DBMS.

- *Grounding on XML*

Not only are the results obtained in XML format, but also the querying instructions are coded in XML. This simplifies the implementation of wrappers for XML/SQL, as it allows XML parsers to be used for parsing queries.

In the present version, XML/SQL does not generate XML attributes. This limitation can be overcome, e.g. by processing XSL transformations on an XML/SQL result.

We are developing a new version of the XML/SQL language which allows XML documents resulting from queries to be altered and returned to the wrapper for it to map document alterations back to the underlying relational database.

We have implemented a wrapper for relational databases which uses the XML/SQL language.

Acknowledgment. This work has been partially supported by IBM do Brasil and Solectron do Brasil.

References

- [Bou00] Ronald Bourret. Xml database products, November 2000.
<http://www.rpbouret.com/xml/XMLDatabaseProds.htm>.
- [BPSM98] T. Bray, J Paoli, and C. Sperberg-McQueen. Extensible markup language (xml) 1.0. W3C Recommendation, February 1998.
<http://www.w3c.org/TR/1998/REC-xml-19980210>.
- [CRF00] D. Chamberlin, J. Robie, and D. Florescu. Quilt: An xml query language for heterogeneous data sources. In *Lecture Notes in Computer Science*, Springer-Verlag, 2000.
<http://www.almaden.ibm.com/cs/people/chamberlin/>.
- [CX00] J. Cheng and J. Xu. Ibm db2 xml extender: An end-to-end solution for storing and retrieving xml documents. ICDE'00 Conference, February 2000.
- [D⁺98] A. Deutsch et al. Xml-ql: A query language for xml. Submission to the World Wide Web Consortium, Aug 1998.
- [FTS00] M. Fernandez, W. Tan, and D. Suciu. Silkroute: Trading between relations and xml. In *Proceedings of Ninth International World Wide Web Conference*, 2000. <http://www.research.att.com/~mff/files/>.
- [I⁺98] H. Ishikama et al. Xql: A Query Language for XML Data. In *The Query Languages Workshop*, 1998.
- [IBM00] IBM Corp., San Jose. *XML Extender: Administration and Programming (Version 7)*, 2000.
- [ISR00] Xml from databases: Odbc2xml, 2000. Intelligent System Research, Chicago, USA.

- [Lad01] R. Laddad. Xml apis for databases: Blend the power of xml and databases using custom sax and dom apis, January 2001.
<http://www.javaworld.com/javaworld/jw-01-2000/jw-01-dbxml.html>.
- [ORA99] Oracle xml sql utility for java. Oracle Corporation, 1999.
http://technet.oracle.com/tech/xml/oracle_xsu/.
- [PGMW95] Y. PAPAKONSTANTINOU, H. Garcia-Molina, and J. Widom. Object exchange across heterogeneous information sources. *IEEE International Conference on Data Engineering*, pages 251–260, March 1995.
- [Vol99] T. Volker. Making legacy data accessible for xml applications, 1999.
- [Wai99] B. Wait. Using xml in oracle database applications, November 1999.
<http://technet.oracle.com/tech/xml/>.

A XML/SQL DTD

The DTD of the XML/SQL language is given below.

```

<!ELEMENT XMLSQL (QUERY,CONSTRUCT)>
<!ATTLIST XMLSQL version CDATA #FIXED "1.0">
<!ELEMENT QUERY (SQL+)>
<!ELEMENT SQL (#PCDATA)>
<!ATTLIST SQL idsql ID #REQUIRED>
<!ELEMENT CONSTRUCT (LIST)>
<!ELEMENT LIST (ATOM|SEQUENCE|LIST)>
<!ATTLIST LIST tagname CDATA #REQUIRED
            idsql IDREF #REQUIRED
            nestby NMTOKENS #IMPLIED>
<!ELEMENT SEQUENCE (ATOM|SEQUENCE|LIST)+>
<!ATTLIST SEQUENCE tagname CDATA #REQUIRED>
<!ELEMENT ATOM EMPTY>
<!ATTLIST ATOM tagname CDATA #REQUIRED
            source CDATA #REQUIRED
            mandatory (no|yes) "no">

```

Composition of XML-Transformations

Johann Eder and Walter Strametz

University of Klagenfurt

Department of Informatics Systems

`eder@isys.uni-klu.ac.at`, `walter.strametz@infologs.com`

Abstract. Electronic commerce seeks improvements of business processes by aggressively exploiting the enormous increases in information exchange offered by digital telecommunication systems. XML is seen as an important step to overcome the problems of heterogeneity of data exchange between different systems, albeit the structural as well as the semantic heterogeneities are not even touched by this standard: The same information is encoded quite differently in XML by different information systems. Therefore, to let these information systems communicate and interoperate, it is necessary to transform XML documents.

We propose a new way to generate such transformations based on the XSLT language which was originally developed for rendering XML documents. We aim to improve the way XSLT transformations are developed by binding XSLT transformers to the document type descriptions of source and target documents and introducing and exploiting the concepts of composition and specialization for DTD as well as for transformers in XSLT, resulting in highly improved efficiency and quality.

Keywords: XML, heterogeneous information systems, e-commerce

1 Introduction

Electronic commerce in all its forms and variants depends on the electronic exchange of information - in interactive form, and/or by exchange of electronic documents. While interactive form is still frequently used in business-to-consumer (B2C) applications, in business-to-business (B2B) applications the exchange of electronic documents is the preferred way. This allows to overcome costly media breaks in business processes and enables IT-systems to interoperate with different IT-systems of business partners [5].

In such a scenario, an IT-system has to be capable of accepting electronic documents from various sources or generating electronic documents for various receivers. The necessity of adapting to formats of electronic documents defined by others depends on the market power of the organizations.

There are no generally accepted standards for electronic business documents, although many attempts have been made (e.g. EDIFACT). Exchange of documents in electronic commerce suffers from heterogeneity on several levels: from the choice of the code, the structure of documents up to semantic differences.

For the lower levels of electronic document exchange XML, the extended markup language [6], is developing as generally accepted standard. It can be soon taken for granted that it is possible to send XML documents to a communication partner and this communication partner is equipped with software to process XML documents. However, it is not at all sure that this communication partner also understands the documents. For successful communication it will be necessary to negotiate the form of XML documents and/or to transform XML documents into different XML representations.

For an example: There are numerous electronic bookshops on the web. So it should be easy to write an application to get the best bid for a given book (including taxes and shipment costs). However, all the book-outlets have different interfaces for costumers and even if a request for quote could be received by sending an XML document, all the document forms are probably different. So for writing the application sketched above, it is indeed necessary to transform the XML document into different forms.

In this paper we report on an approach to facilitate the development of XML-transformers. We use the widely available language XSLT [3,7,6,8,11] as transformation language. To overcome some shortcomings of this language we first bind XSLT transformers to the DTD (document type definition) of the source and destination document types. This makes it easier to search for appropriate transformers when a new one has to be developed. Then we introduce the notions of composition and specialization of XML documents as well as of XSLT transformers and provide a meta structure for storing this information. Then we can use this structure for composing new XSLT transformations from already available component transformers.

We will also briefly describe our prototype implementation of a transformation system named CoX (component-based XML transformer), based on these concepts.

2 XML and XSLT

In this section we revisit the basic notions of XML and XSLT to introduce the concepts and terminology we need in the following sections to make the paper more self-contained.

2.1 XML and DTDs

Semi-structured data [2,19] can be represented with the XML language [3,10] standardized by the W3C. Semi-structured data (documents) are modelled in form of trees, whereby nodes contain data and the named edges (*tags*) describe the nodes (*elements*). The labels are interpreted as schema information and thus the tree contains the schema information of the document. For an example, Figure 1 shows a sample document of a simplified order document.

The tree-representation of an XML document such as an order can be constrained and further documented by a *Document Type Definition* (DTD), which

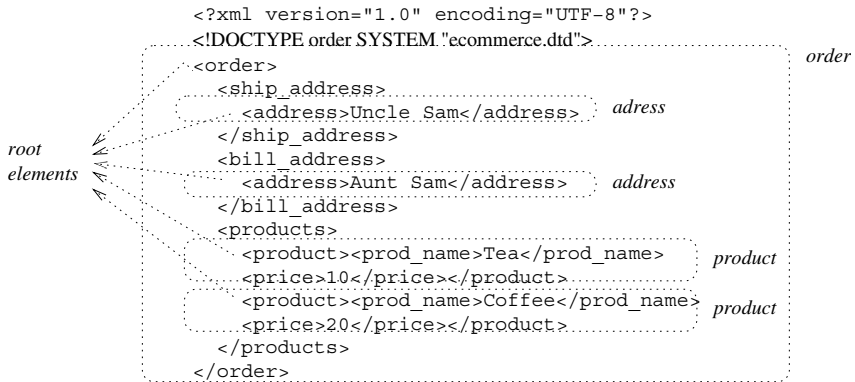


Fig. 1. An order-document in XML. The dotted rectangles indicate the components of the document and the arrows the root elements of the components.

is part of the XML language. In Figure 2 the DTD of the order document of Figure 1 is shown. A DTD is a context-free grammar for the structure of an XML document. An XML parser validates the conformance of a document with a DTD. For example, in the order-document a product-tag inside an address-tag should not be accepted. The DTD is an optional part of an XML document but in this paper we only consider documents with DTD - syntactically expressed in the DOCTYPE clause of an XML document. E.g. the clause `<!DOCTYPE order SYSTEM "ecommerce.dtd">` binds the DTD "ecommerce.dtd" to the document of Figure 1.

```
<!ELEMENT order (ship_address, bill_address?, products)>
<!ELEMENT bill (bill_address, products)>
<!ELEMENT ship_address (address)>
<!ELEMENT bill_address (address)>
<!ELEMENT address ANY>
<!ELEMENT products (product)*>
<!ELEMENT product (ANY)>
```

Fig. 2. The DTD of the XML document in Figure 1.

As a DTD may feature many choices and optional parts, a particular DTD can describe several disjoint sets of document schemas - for example it is possible that one single DTD can validate an order-document as well as an invoice. The DTD in Figure 2 uses this property to validate an order as well as a billing document. Thereby, the root-element of the document is used as an entry-point to the DTD to put together the grammar for validating the schema-tree. As

you can see the `bill` element shares structure (`bill.address`, `products`) with the `order` element. So parts of the DTD can be reused with the consequence that documents share structural parts for particular subtrees of the XML document. The main idea of our approach is to reuse the transformations of shared structures among different documents which are based on the same DTD.

2.2 Transforming Documents with XSLT

XSLT (*eXtensible Stylesheet Language for Transformations*) [3,7,6,8,11] is a language for transforming XML documents. A transformation in the XSLT language is expressed as an XML document and can therefore be validated and parsed by XML parsers. A transformation expressed in XSLT consists of rules for transforming a source tree of an XML document into a result tree, which are executed by an XSLT processor. The XSLT standard does not specify how an XSLT transformation is associated with an XML document. As a consequence a system has to keep track which transformations must be applied on a document in order to get the desired output. When transforming various document types this task becomes difficult to manage.

As a solution to this we wanted a transformation system which supports document types: A transformer (consisting of a set of XSLT transformation programs) should be simply invoked by providing the source document and a target type of the output document. The transformation process itself should be transparent to the user. Our solution to this problem is to bind the XSLT transformation to the DTDs of source and target documents. The CoX transformation system uses this information to find and apply the proper transformers for the specific instance of a document type. We extended this basic idea and added support for composition and specialization to further simplify and accelerate the process of creating document transformations.

3 Composition and Specialization of Documents

3.1 XML Components

A document can be seen as an aggregation of components. For example, in a document for ordering goods we can identify the shipping and billing address, the products, etc. as components of the ordering document. In an XML document the tags can be interpreted as delimiter of the components of the document. In the tree model thus we define a component of an XML document as a subtree identified through its root element.

The type of a component is then defined as the elements reachable from the type name of its root element in the DTD of the document. A component can in this way be validated against its DTD and type using conventional XML parsers.

In principle, every element defined in a DTD can be considered as a component. However, we recommend to restrict to those elements which represent semantically meaningful units (entities) of the problem domain.

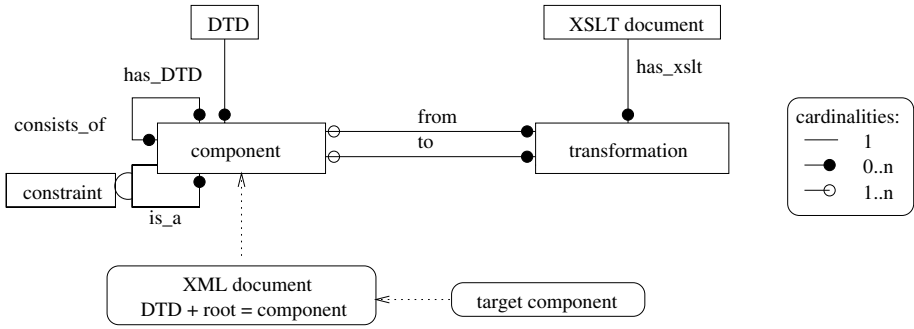


Fig. 3. Type information stored within the CoX transformation system (metaschema).

In the metaschema we represent the aggregation hierarchies of documents and of document types. Fig. 3 shows the realization of the definitions above with the associations from component to the entities DTD and root: A component is related to exactly one DTD and has exactly one root element. Defining components only makes sense if there are several components for one DTD. Later, when we introduce specialization it will be clear that having multiple components which have the same DTD and the same root element is perfectly reasonable.

We can generalize the meaning of a component in a way that also a whole document can be seen as a component. So a document is a component with a certain base type (i.e. a DTD together with a root element). Like a document can have components also a component may be composed of components in turn. The *consists_of*-relation of Figure 3 reflects that a component may consist of several components. The model also shows that every component is assigned to a DTD and to an element name. Given that information an XML parser can validate a component.

3.2 Specialization of XML-Components

So far we presented the aggregation structure of XML documents and XML types. In good modeling tradition, we also introduce the notion of specialization. This concepts covers three different phenomena:

1. specialization through environment
2. specialization through restricted structure
3. specialization through restricted instances

The first type of specialization allows to distinguish between components with the same type but different semantics due to different environment. As an example for this kind of specialization we look at the components shipping and billing address of the example above. Both are addresses and we can't distinguish

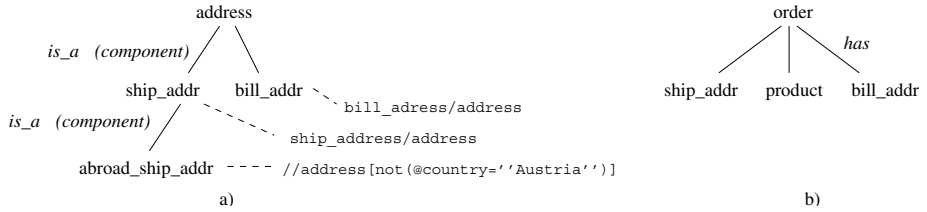


Fig. 4. a) is an example of a specialization tree of the address component. b) shows the component tree of an order document.

between them by type and therefore we can bind only one transformation to the addresses.

The second kind of specialization considers the fact that a document type definition may be very generic and allows the validation of many different structures of documents. We define a specialization as a DTD which restricts the structural genericity of a DTD. For an example, it might be required that the shipping address is a physical address with street and number and not merely a post-box address. So the physical address restricts the DTD of address and it would validate only a subset of valid address documents.

The third way of specialization restricts the possible instances of a type by restricting values. As an example, certain values can be defined as mandatory. The type "domestic-address" would require that the country component of an address is always filled with Austria, in our example.

For all three kinds of specialization we found a uniform way of representing them in our metaschema: We define a component *A* as specialization of component *B*, if every document which is valid for *A* is also valid for *B*, and additionally satisfies the specialization constraint. The relationships between components defined through specialization is represented through the *is_a* hierarchy in our metaschema where the condition attribute of the *is_a* relationship represents the specialization constraint.

The specialization constraints are expressed in the language XPath [3,12] which is part of XSLT and thus can be processed by an XSLT processor. XPath provides a way to select a subtree of a document by selecting a specific element. If the selected subtree is identical to the instance of the specialized component the constraint is satisfied. The identity is checked by comparing the root element of the component instance with the result of the XPath expression. If both elements are equal then the test succeeds. The test is also called an "element test".

XPath allows to specify the path to an element of a document. The syntax of a sample XPath constraint for the shipping address is: "**ship_address/address**" which matches any **address** element with an **ship_address** parent (see also Figure 4a). With this functionality it is possible to distinguish between the billing and the shipping address of the sample document in Figure 1. When the transformer recognizes an address it executes an element test for the root element (**<address>**) of the component. It evaluates the XPath expression which is as-

signed to a specialized component. If the result of the XPath expression contains the root element of the component instance then the test is successful and we can infer that the document is valid for the specialization (i.e. the actual type of the document is the subtype considered). [7,6,12] provide details of XPath and the expressive power of this language.

3.3 Specialization and Transformation

The specialization allows to provide different transformations for documents with the same DTD based on the their dynamic subtypes, i.e. on the specialization whose specialization conditions they satisfy.

In our metaschema (Figure 3) the *is_a* relationship represents the specialization for components. A component can be a specialization of an other component, whereby all assoziations and attributes are inherited from the parent. The assoziations except *has_DTD* can be overwritten by the specializing component. Thus the base type of a specialization hierarchy is the same for all components.

Figure 4a shows an example of the specialization hierarchy of the address component and its constraints. The root of the tree is the most general component with the most general transformation. Each specialization adds a constraint to the constraints inherited by its preceding components. To meet the criteria for a specialized component all constraints must be checked successfully. In the example the *abroad_ship_addr* component has to satisfy two constraints - it is a shipping address (environment specialization) and it does not contain country code "Austria" (instance specialization). The transformation of this component can be highly specialized in transforming "abroad shipping addresses" and the transformation program can rely on the constraints defined in the specialization hierarchy. If no transformation is defined for this component then the transformation of the *ship_addr* component will be applied. It is guaranteed that all possible specializations of *ship_addr* meet the criteria to be a *ship_addr* component and, therefore, all transformations defined for *ship_addr* can be applied.

4 Generating Transformations

4.1 The Process of Transforming a Document

In this section we discuss how a document is transformed with the CoX transformation system. Assume that some XSLT transformations are already programmed and the type information is stored within the system. We start with an XML-document and the target type. With the DTD and the root element of the document the principal component can be found. Now the type of the source and the type target document is known. The CoX transformer will only apply transformations which lead to the DTD of the target component.

To determine the dynamic type of the document the transformer analyzes the source document and determines the base type by using the DTD and the root element of the component. The *is_a* relation is searched depth-first to apply

the constraint expressions on the document. So a document is decomposed and the most specialized subcomponents are identified by checking the constraints given in the specialization hierarchy.

When a component is transformed it is important that the transformation has access to the component only. Otherwise an XSLT transformation could have undesirable side effects on the whole document. To obtain a scope for each component the CoX transformer extracts the XML subtree representing the from the document. Each component is transformed as it was a separate document.

For the transformation, therefore the document is decomposed into it's components which are in turn transformed and afterwards (re-)assembled to obtain the target XML document.

The transformer uses the *consists_of* association to recursively determine the smallest components for which a transformation program to the target DTD has been registered. For each of the components identified in this way, the *is_a* hierarchy is descended to identify the most specialized dynamic type of each component in a depth-first manner. A component is thus transformed if its *consists_of* relation is empty or if all of its components are already transformed. If no transformation is found the search continues at the next higher component in the specialization hierarchy.

4.2 Transformation of a Sample Document

On the sample document of Figure 1 we want to show the principle of how a transformation of a document is generated.

First, the transformation system has to make sure that the schema information is correct. For example the component and specialization trees must be checked if they are correct trees and have no circular links. Another requirement is that all components in the component tree have the same DTD, and so on. The data for the metaschema of Figure 3 is stored in an external XML document which is read at startup time of the transformation system.

With the CoX transformation system it is possible to transform one document to many other types and DTDs simply by naming the target component or target DTD. In the following example a document which represents an order document (see Figure 1) should be transformed into an HTML representation. In Figure 4b you can see a fragment of the type information of the order document which is stored within the CoX transformation system. The specialization tree of the address component (see Figure 4a) is also stored within the system. For the transformation of the components there are five transformations defined¹: *t_order*, *t_addr*, *t_ship_addr*, and *t_prod* which are assigned to the according components. Note that there is no transformation for the billing address.

With the DTD and the root element of the document the transformation system selects the proper component tree of the order document. From there the transformer learns to look for an **address** component. Two address components

¹ The transformations may be implemented with one XSLT document or even more than five documents using **import** and **include** statements.

are found which are processed consecutively. As there are specializations for an address it searches the specialization tree, evaluates the XPath constraints and determines the dynamic subtype, which is *ship_addr*. This leads to the *t_ship_addr* transformation which is applied to the subtree (starting at the first `<address>` tag). The second address is processed similar but the dynamic subtype is *bill_addr*. This component has no transformation assigned and therefore the transformation is taken from the *address* component. In other words the billing address inherits the transformation from the address component. Then the transformations for the products are applied. The final transformation is the order transformation which is the only transformation having access to the whole document. This way also a final rearrangement or the merging of components can be carried out. So the transformer executes the following sequence of XSLT transformations in their particular scope: *t_ship_addr*, *t_addr*, *t_prod* and *t_order*. The transformation of the whole document is composed of a set of transformers for parts of this document and can be generated from these partial transformers.

4.3 Creating New Transformations

The more transformations and different DTDs are employed in a system the more important it gets to manage the transformations. The CoX transformer helps to organize documents and transformations in reusable pieces. A major goal for the development of CoX was to support the programmers of transformation programs to increase the efficiency of the process for creating transformations and to achieve quality improvements by employing certified well tested component transformers.

The process of creating a new transformation starts with searching the meta-structure whether transformations for the DTD of the document have been defined already. If no suitable transformation is found, the search is continued for the components of the DTD. If transformation programs for components are found, they can be readily used as part of the transformation of the new document. The specialization hierarchy also supports the creation and application of transformations which reflect the actual structure (dynamic type) and content of a document, which increases the flexibility for handling semi-structured data in a flexible yet reliable way.

4.4 Implementation

The previous sections introduced the concepts of composition and specialization for XML transformations. The ideas are applicable for the transformation of an XML document and do not depend on any transformation language. To keep the implementation simple we have chosen XSLT as the transforming language. The CoX transformer is written in the Java language and is based on the XSLT/XPath processor "Xalan" which is available on <http://www.apache.org>. The architecture of CoX makes extensive usage of the factory pattern [4] which

makes it possible to exchange the XSLT and the XPath processors. The type information about components, DTDs and specialization trees is stored in an XML document which is parsed and evaluated by the CoX transformer. For that it uses the "Xerces" XML parser which is also exchangeable. The transformation system is accessible via a Java, a commandline and a graphical interface.

5 Conclusions

We presented the XML transformation systems CoX and its underlying methodology. CoX was developed to support the process of developing transformations of XML documents from one format to another. The main contributions of this approach are the following:

- Recording source and target DTDs of XSLT transformations.
- Composition and specialization of XML documents.
- Composition and specialization of XSLT transformations.

The information obtained by decomposition and specialization is documented in a metastructure which is then used in the process of identifying available transformations.

The methodology introduced in the CoX system is intended to increase the efficiency of the development process for creating transformations between XML documents. It greatly increases the productivity by promoting reuse of already developed transformations (for components) and thus reduces the necessity of developing new transformations from scratch.

References

- [1] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web*. Morgan Kaufmann Publishers, 2000.
- [2] P. Buneman. Semistructured data. Tutorial in Proceedings of the 16th ACM Symposium on Principles of Database Systems, 1997.
- [3] World Wide Web Consortium. W3C. <http://www.w3c.org/>, 2001.
- [4] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Entwurfsmuster - Elemente wiederverwendbarer objektorientierter Software*. Addison-Wesley, 1996.
- [5] H. Groiss and J. Eder. Workflow systems for inter-organizational business processes. *ACM SIGGROUP Bulletin*, Dec. 1997.
- [6] E. R. Harold. *XML Bible*. IDG Books Worldwide, 1999.
- [7] D. Martin, M. Birbeck, M. Kay, et al. *Professional XML*. Wrox Press, 2000.
- [8] T. Milo, D. Suciu, and V. Vianu. Typechecking for XML transformers. In *Proceedings of the Nineteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2000, Dallas, Texas)*, 2000.
- [9] D. Suciu. An overview of semistructured data. *ACM SIGACT News*, 1998.
- [10] D. Suciu. Semistructured data and XML. In Proceedings of International Conference on Foundations of Data Organization, Kobe, Japan, 1998.
- [11] P. Wadler. A formal semantics of patterns in XSLT. Markup Technologies, 1999.
- [12] P. Wadler. Two semantics for XPath. <http://cm.bell-labs.com/cm/cs/who/wadler/>, 2000.

Classification and Characteristics of Electronic Payment Systems

Dennis Abrazhevich

IPO, Center for User-System Interaction
Technical University of Eindhoven (TUE)
PO Box 513, 5600 MB
Eindhoven, The Netherlands
D.Abrazhevich@tue.nl

Abstract. Electronic commerce and electronic business greatly need new payment systems that will support their further development. To better understand problems and perspectives of the electronic payment systems this article describes a classification and different characteristic aspects of payment systems. It suggests distinctions between payment systems and mediating systems, and is trying to illustrate advantages and limitations of diverse categories of payment systems using the defined characteristics. It is highlighting importance of user-related aspects in design and introduction of electronic payment systems for mass customers.

1 Introduction

The explosive development of electronic commerce in recent years makes the issue of paying over open networks very important. Electronic payment systems are required to bring the necessary infrastructure to facilitate payment over the Internet. They are becoming an essential part of, and are greatly necessary for, further development of electronic commerce and electronic business.

The problem that we are facing at present is that the conventional ways of paying for goods and services do not work suitably over the Internet. Existing payment systems for the real world, such as credit cards are widely accepted as means of payment on the Internet, however their use encounters difficulties among users who do not see in them enough trust and security, [11]. The existing payment systems are also far from ideal for merchants, because of the high transaction costs, fraudulent activity and the multiple parties involved in payment processing [16]. These problems result in reluctant participation of users in e-commerce activities that involve paying over the Internet, and this situation in turn affects merchants who are losing potential customers. The need for well-performing and user-friendly payment systems that would satisfy both sides emerges clearly. These systems should meet needs of users and merchants, and demonstrate a potential for acceptance on a mass market scale.

As an example of the problems that designers can face we can look at what happened to the Chipknip and Chipper systems in the Netherlands. These are smart card-based systems; to use them, a customer has to charge a smart card with money in advance. These systems initially were designed with the intention to provide another way of paying for small purchases, reducing the necessity to pay with cash. In spite of constant improvements of the technology these systems still encounter low level of

acceptance from mass customers: only 1/3 of the 15 m. of issued Chipknip cards are loaded with money, according to Interpay (the operator of the Chipknip system), which is quite few. Despite significant efforts to promote the technology Chipknip and Chipper have not received much popularity until now [4]. Finally, Chipper was closed down in 2001 [9]. The opinion of experts in this field is that the main problems were lack of focus on end users, lack of applications and places where one can pay with the card, and incompatible standards of the systems.

The history of credit cards, on the other hand, demonstrates different problems. Though being broadly used in many applications and even competing with cash in some countries, credit cards could not avoid the reputation of an insecure and untrustworthy payment method. This is happening due to continuous issues of fraud and counterfeiting of credit cards that result in money losses for cards owners, banks, vendors, piling up to huge numbers of credit cards frauds every year. Therefore it is no surprise that this results in low trust of vendors and consumers in credit cards as a payment method [16].

This article presents classification and characteristics of payment systems with the aim to better understand the problems and possible ways of future development. Technical realization and successful implementation are the fundamental issues to be solved, but even if there are good technological solutions and they are not accepted by end users or vendors, the whole system would fail. Therefore, in order to design a payment system that is successful, in the sense that it is welcomed by users, one should pay attention to the variety of aspects of electronic payment systems, which are described in this article. This article aims to highlight what aspects of payment systems are important from the point of view of the interested parties, especially for the end users.

2 Classifications of Payment Mechanisms: State of the Art

Before discussing problems, limitations and success criteria of electronic payment systems a classification is presented below to better understand the field and identify characteristic properties of the systems.

2.1 Basic Classification: Account- and Token-Based Mechanisms

The first level in the categorization is based on the way in which money transfer is organized. Existing payment mechanisms may be divided into two groups: electronic currency systems (or electronic cash) and credit-debit systems [12].

Another terminological approach offered by Wayner [17], based on the type of information that is exchanged, distinguishes between “account-based” and “token-based” systems, which respectively correspond to electronic currency and credit-debit systems, according to the definition above. This terminology describes the distinctions between the systems more accurately and therefore we will use it in this paper.

Electronic currency resembles conventional cash, when parties exchange electronic tokens that represent value, just as banknotes determine the value of paper money. The credit-debit approach in the context of electronic payments means that money is represented by numbers in bank accounts and these numbers are transferred between parties in an electronic manner over computer networks. Some writers put credit cards

systems in a separate group [12]; others consider them to be a variant of credit-debit system.

Going one step further in the classification in the group of account-based systems we can distinguish between debit and credit cards systems and specialized ones, e.g. those systems that use e-mail for money transfer or notification. Electronic currency, in its turn, can be divided on systems that support smart cards, and those that exist only in online environment. They can be called ‘online cash’ or ‘Web cash’. Prepaid card and electronic purse systems can be also included in this category.

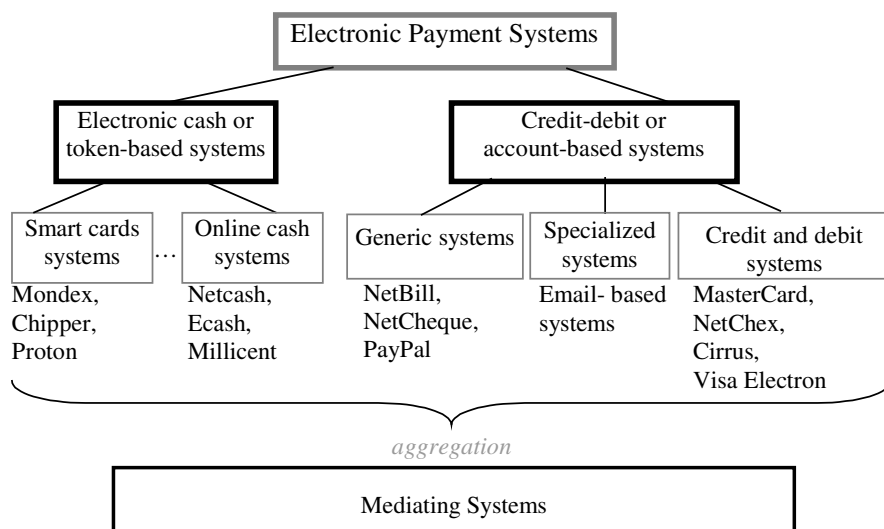


Fig. 1. Classifications of electronic payment systems

2.2 Mediating Systems vs. Payment Systems

Proceeding with the classification we have to make one important distinction, which is between *mediating* and payment systems. This partition makes sense particularly in the context of electronic and Internet payment mechanisms. Mediating systems appeared to overcome the imperfection and inefficiency of current payment systems for the Internet. Number of solutions has been proposed as solutions to this problems, extending the services of the existing systems and providing a medium that would facilitate paying over the Internet. In mediating systems existing infrastructure is aggregated to provide better services. Besides, a special class of mediating systems has emerged with an intention to provide a convenient service for paying bills, thus facilitating e-business practices. Providers of *billing systems* target mainly businesses although some of them provide bill payment services for end users.

From this we can draw several examples of mediating systems. The obvious one is the systems that provide bill payments for companies and end users. Bills that normally are paid in non-online ways as paper cheques or bank transfers can be paid over a Web front-end provided by the billing systems.

Another example of mediating systems is solutions that aggregate existing payment mechanisms and give an opportunity to make and accept payments via a Web site. The payment process in this case is transparent to users of the site, who perceive that they are paying at it, while in reality a mediating service provider ‘intercepts’ payments from users, processes them and charges accounts of the owner of the site when the authorization and transactions are completed. Among mediating services providers there are numerous credit cards processing companies.

Described payment mechanisms are essentially mediating systems, rather than payment, because they only extend existing infrastructure so customers can use their money to pay over the Internet. The difference between mediating and payment systems can be summarized in that mediating systems serve as a *mediator* between payer, business and payment systems, while there is no such mediator in for payment systems.

A good illustration of a mediating system is a payment service of Bibit Billing Services, [2]. The company specializes in Internet payments and billing services. The service supports about 65 payment methods from 14 countries. Processing of the transactions for selected payment system lies entirely on Bibit and is transparent to the visitors of the site and the client company. The company business model relies on providing extra services to facilitate payments. Other examples of mediating systems are PayTrust (www.paytrust.com), PayNet (www.paynet.ch), iBill (Ibill.com) and more.

This article is mainly focused on payment systems, not mediating solutions for existing payment infrastructure that work, for instance, for the Internet. The reason is that mediating systems on the Internet emerged because of the absence of relevant *payment* solutions. Many of them are probably temporary systems that are not able to completely resolve the problems that appear in the Internet context, because the problems reside in the payment systems they use. We discuss these problems later in the text.

2.3 Identifying Characteristics of Payment Systems

As we observed on the example with Chipknip and Chipper in the previous section there are lot of factors that determine success or failure of payment systems, and not all of them are of technical nature. Users acceptance depends on many user-related issues, such as consumer choice, preferences, state of the market, etc.

There are several attempts to describe characteristics of payment systems, mainly from technological side [12, 1]. If we want to better understand how payment systems are perceived by users and what problems they have, we can define characteristics that describe these systems from various points of view as technological, user-related, market, legal and other issues.

Descriptions of the Characteristics of Payment Systems. The discussion of diverse aspects of electronic payment systems can be found in many works on the development and research in the field. Attempts to classify and provide descriptions of characteristics of payment systems such as trust, ease of use, security, reliability, flexibility, convertibility, efficiency, traceability and other can be found in [11, 12]. Charac-

teristics of payment systems found in the literature are: anonymity, applicability, authorization type, convertibility, ease of use (usability), efficiency, interoperability, reliability, scalability, security, traceability, trust. We have to note however, that these works are mainly focused on technical aspects of electronic payment systems, which is only one of the facets of payment systems.

In many cases users can immediately perceive diverse aspects of payment systems when interaction takes place, e.g. applicability, convertibility. However, there are other characteristics that are experienced indirectly, but still have influence on users.

To illustrate this difference the described characteristics can be divided on those that are perceived by users directly and those that are 'transparent' to users.

Direct perception (user related characteristics): anonymity, applicability, convertibility, ease of use (usability), efficiency, reliability, security, trust, traceability.

Indirect influence (technology related characteristics): scalability, divisibility, interoperability, authorization type. Further research will investigate that this division is reasonable in respect that users really do not have understanding and direct experience with 'indirect influence' characteristics, so this would not be only an assumption. The next step will be to concentrate on user related characteristics because they have immediate influence on users and, therefore, on user acceptance of electronic payment systems.

The characteristics of payment systems are extended below to account for various aspects such as user-related, political and technological factors. They can also be transformed into requirements for a future system.

Applicability. The usefulness of a payment mechanism is dependent upon what one can buy with it. **Applicability** (or acceptability, as it often mentioned in literature) of a payment system is defined as the extent to which it is accepted for payments. For instance, cash is accepted widely and thus has high level of applicability. Applicability of a payment system may vary from country to country. Quite high applicability have debit cards (bankcards) and credit cards, while cheques are not longer common in several European countries.

Ease of use. Paying with an electronic payment system should not be a complex task; we will call this characteristic **ease of use** or **usability**. Usability is an important characteristic and defined as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use", [7]. Payments should be automated and done in an easy, seamless way. In such a responsible task as a payment process users should have minimum factors that make it difficult to pay or distract them.

Security. Since Internet services are provided today on the open networks, the infrastructure supporting electronic commerce, and payment systems in particular, must be resistant to attacks in the Internet environment. For electronic cash systems the issue of security has a special angle of counterfeiting, which means that no one should be able to produce electronic tokens on their own. Another angle is double spending; design should ensure that electronic tokens couldn't be spent twice.

Reliability. Naturally, users would like to see that system is **reliable**, because the smooth running of an enterprise will depend on the availability of the payment infrastructure, [12].

Trust. Note that proper manifestation of the most characteristics brings to the system the vital attribute of **trust** [6]. Trust in this context refers to the degree of confidence that money and personal information will be safe and that parties involved will

not act against user's interest. From the perspective of using a payment system people trust that payments will be conducted in a proper way, that money will not be stolen or misused. On the other hand, even if we use an imperfect system we believe that vendors, banks and credit cards companies will not use the information against us. Another aspect of trust is that other parties should have trust in the payment systems we want use, based on this trust they would be willing to conduct commerce.

Scalability. As the commercial use of the Internet grows, the demands placed on payment infrastructure will also increase. The payment infrastructure as a whole should be **scalable**, to be able to handle the addition of users and merchants, so that systems will perform normally without performance degradation, [12]. Among the least scalable systems are those that require from users and vendors purchase and installation of additional hardware; this often hampers development of electronic cash systems.

Convertibility. Naturally, users will select payment mechanisms as financial instruments according to their needs. Numerous payment schemes have emerged up to now and we can expect new systems to appear, all providing assorted services and applications for assorted users needs. Funds represented by one mechanism should be easily **convertible** into funds represented by others.

Interoperability. A payment system is **interoperable** if it is not dependent on one organization, but is open and allows as many as necessary interested parties to join. This can be achieved by means of open standards for the technology that is used. It is natural, though, that companies that implement new technologies treat them as know-how, because of the added value they create by investing in the technologies; therefore it is not always sensible to demand interoperability. Examples of interoperable initiatives are the CAFE project, [3] and SEMPER project (www.semper.org).

Efficiency. There are lots of discussions on the ability of payment systems to accept "micropayments". Small payments are amounts less than a \$1; micropayments are less than several cents. Systems should be able to receive small payments without performance degradation and posing high costs of transactions, [10]. The costs per transaction should be reasonable for processing small amounts.

Anonymity. The following characteristics reflect a desire to protect one's privacy, identity and personal information. For some transactions, identities of the parties in the transactions could be protected by **anonymity**. Anonymity suggests that it is not possible to discover someone's identity or to monitor an individual's spending patterns. Where anonymity is important, the cost of tracking a transaction should outweigh the value of the information that can be obtained by doing so. For an illustration, when one pays with a debit card, the purchase is registered at vendor and bank's databases. It is possible to find out what amount was paid and what actually was purchased, thus debit card payments are not anonymous. On the contrary, we can state that cash is an *anonymous* payment system, as there is no direct information about you personally on banknotes. There are laws in several EU countries that limit usage of personal information by banks, authorities and other parties.

Traceability. Anonymity relates to the characteristic of **traceability**. Traceability indicated how easy it is to trace money flows, sources of funds or link spent funds to a customer via payment activities. There are lots of films where police is able to find someone who is using credit cards in any place over the US and even worldwide. This suggests that credit cards are traceable. Anonymity and traceability are important for users as characteristics that induce **trust**.

Authorization type. Another characteristic of this kind is **authorization type** that is addressed in the literature as an ability of a system to perform payments not being connected to a central authority, [1, 11]. Authorization type can be offline or online. Offline authorization type means that users of the system can exchange money not being connected to a network, without a third party as a mediator. Natural illustration for offline authorization is the exchange of cash. Some electronic payment systems, e.g. Mondex, also offer this kind of service.

3 Examples of Categories of Payment Systems

Having described the various characteristics of payment systems, we will discuss the advantages and limitations of different categories of payment systems with their assistance.

3.1 Electronic Currency or Token-Based Systems

When using electronic currency systems customers purchase electronic digital tokens from the issuing company. Electronic currency represents value in some form and can be spent with merchants, who deposit the currency in their own accounts or can spend the currency in other places. Electronic currency is stored in digital form and serves as a cash substitute for the Internet. It can be represented by electronic “bills and coins”, certificates, packets of data, or, in other words, tokens.

Customers can pay for tokens by using credit cards, electronic checks, or other ways, if possible. Some of the systems allow converting electronic currency back in another form of money, [12]. Examples of electronic currency systems are e-Cash, based on the payment system invented by David Chaum [5], the NetCash system [12], Mondex, PayWord and MicroMint [13].

The further partition of the electronic cash systems is between those that use smart cards for storage of tokens and those where tokens reside only on user's accounts and computer networks. Later can be addressed as ‘Web cash’ systems. This means that tokens of these systems live and travel in computer environment, for example on a currency server or user's hard disk. Web cash representatives are e-Cash, NetCash, DigiGold, Milicent, PayWord and MicroMint.

Systems that employ smart cards like Chipknip, Chipper, Belgium Proton, Mondex and Visa Cash can be also placed in the category of electronic cash, although by the way of representing money they hardly use any tokens, but rather a number stored on the card is changed when a payment takes place. By the principle of work they act like an electronic purse. The value is stored on a card and if the card is lost the money is gone, in a fashion similar to cash. This gives the opportunity to situate these systems in the “electronic cash” branch of the classification.

A prominent smart card payment system is Mondex, which was designed to enable person-to-person as well as Internet payments. The card can be used for small payments, store personal and application specific information and serve as a telephone card. The level of service of Mondex is very high. This card demonstrates an example of a solution of good applicability.

As an advantage of electronic currency systems we can note the possibility of payer-to-receiver exchange without the need to contact a central control system. This

can reduce the cost of transactions and facilitate micropayments, system becomes more efficient.

An important advantage of electronic currency is its potential for anonymity. Some systems, like eCash, can block attempts to identify the user to which a specific token was issued even if all parties conspire. In a case of double spending for *offline* electronic cash, if a user is attempting to spend the same tokens twice, the system reveals enough information to determine his identity. NetCash provides weaker form of anonymity, however a user can choose the currency server and can select one it trusts not to retain information needed to track such transactions. Despite Mondex being an electronic cash system it is not anonymous, because each card has a unique identification number that is linked to the person to whom the card was issued at the bank. A user cannot buy a Mondex card without revealing their identities, [8].

A significant disadvantage of current electronic cash mechanisms is the need to maintain a large database of past transactions to prevent double spending. For example, in eCash it was necessary to track all certificates that have been deposited. With the NetCash approach, it is needed to keep track of all certificates that have been issued, but not yet deposited. This can be an obstacle for system expansion, because it can reduce the scalability of the system, [12].

Another factor that may be perceived as a disadvantage is the necessity to purchase and install extra hardware and software, sometimes for both merchants and customers. While for consumers it can mean scrutiny with technical issues and learning a new system, for merchants it may suggest even more efforts for integrating new systems in their accounting and financial reporting. This can also lower scalability of electronic cash systems. However, dedicated hardware may help to solve different problems with security and authentication over open networks.

3.2 Credit-Debit Instruments or Account-Based Systems

In the basic principle of work of account-based systems lies establishing of accounts with payment service providers. Users can authorize charges against those accounts, as they would do with usual accounts. With the debit approach, the customer maintains a positive balance of the account and money is subtracted when a debit transaction is performed. With the credit approach, charges are posted against the customer's account and the customer is billed for this amount later or subsequently pays the balance of the account to the payment service.

One of the most widely used systems for electronic payment, the debit card, is a clear example of debit systems. Other payment mechanisms that use the credit-debit model are NetBill [14], First Virtual's InfoCommerce system, NetCheque system and many more.

A special group of account-based instruments that are currently in wide use employ credit cards systems. A majority of trade on the Internet is done using credit cards and this payment system should not be overlooked. Numerous mediating systems for network payments employ this mechanism. The biggest advantage of this approach is that the customer does not necessarily need to be registered with a payment service; all that is needed is a credit card number. This also results in high scalability, as no additional installations are necessary. Credit cards provide a large customer base for merchants who accept them, thus acceptability is very high. When using credit cards over open networks, encryption mechanisms, such as Secure Socket Layer, in princi-

ple can prevent a hacker or eavesdropper from intercepting the customer's credit card number. There are some schemes that even hide card numbers from the merchant, providing protection against intercepting cards' details from merchant databases or against fraud by the merchant. Nevertheless, these incidents happen regularly, [15].

It is important to note, however, that without some form of registration of customers with a payment service, which means providing substantial proofs of one's identity, credit cards can be very insecure to pay with. As long as even encrypted Internet credit card transaction does not include a signature, anyone with knowledge of the customer's credit card number and expiration date can create a payment order. Also, because online payments are administered as standard credit card charges, costs are high enough to make this method not suitable for micropayments, thus it is not efficient. Credit card companies have been constantly lowering the minimum amount that can be paid, hence small payments can be performed, but charges for merchants still remain high.

Advantages of the credit-debit model are its ease of use and scalability. As long as it presumes using the existing networks and a computer as a payment terminal, there is no need for creating new hardware or infrastructure. Systems built by this model have the potential for good scalability, which allows more users to join the system without great loss of performance. The reason is that to support more users a system should only increase number of accounts that can be done relatively easy; there is no need to support large databases tracking all issued tokens as in electronic currency systems.

There are several limitations in these types of systems. They are usually traceable and not anonymous, so one's spendings and money flows can be easily observed. Account management is usually under the control of the company that provides this service; this can affect reliability (as long as this company has a single point of failure) and interoperability (if it is difficult for other parties to join due to close standards). These types of systems usually require a network connection and do not offer possibilities of offline payments, which is also a limitation in certain contexts of use.

This section has defined the characteristics and classification of payment systems. This research aims to locate problems as they are experienced by users and will define the ways in which user acceptance and, consequently, success of new systems can be improved. Future steps will propose how to operationalize and the characteristics, thus acquiring necessary knowledge for future design.

4 Conclusions

To better describe payment systems from different aspects and be able to highlight the problems and requirements we have discussed classification and an extensive set of characteristics of electronic payment systems. Importance of user-related characteristics of electronic payment systems has been highlighted. Ongoing work is concerned with finding out how users judge the relative importance of characteristics for user acceptance. As a possible future step a system may be designed and tested with users' involvement. This will increase the chance of coming up with valid recommendations for the design of payment systems, which will probably result in better user acceptance.

References

1. Asokan, N., Janson, P.A., Steiner, M and Waidner, M.: The State of the Art in Electronic Payment Systems. *IEEE Computer*, (1997) 28-35
2. Bibit Internet Payment Services, <http://www.bibit.com> (2000)
3. Boly, J-P. et al.: The ESPRIT Project CAFE. *ESORICS 94, LNCS vol. 875*, Springer-Verlag, Berlin (1994) 217-230
4. ACI Adds Chipper To Its Chip Portfolio, *Card Technology*, June 2000, <http://www.cardtech.faulknergray.com/jun00.htm> (2000)
5. Chaum, D.: Achieving electronic privacy. *Scientific American*, vol. 267, no.2, (1992) 96-101
6. Egger, F.N.: Trust Me, I'm an Online Vendor. *CHI2000 Extended Abstracts: Conference on Human Factors in Computing Systems*, The Netherlands (2000)
7. ISO/DIS 9241-11.2, Part 11: Guidance on usability specification and measures, Part of: ISO 9241 (1996)
8. Jones, D.: Mondex: A house of smart-cards?. *The Convergence*, <http://www.efc.ca/pages/media/convergence.12jul97.html> (1997)
9. Libbenga, J.: Ondergang Chipper markeert nieuw begin. (Fallen Chipper marks a new start), *Automatisering Gids* No. 10, the Netherlands, March 9 (2001)
10. Low, S. H., Maxemchuk, N. F. and Paul, S.: Anonymous credit cards. In *Proceedings of second ACM Conference on Computer and Communication security*. (1994) 108-117
11. Lynch, D.C. and Lundquist, L.: *Digital money: The new era of Internet commerce*. Chichester: Wiley (1996)
12. Medvinsky, G. and Neuman, B.C. Netcash: A design for practical electronic currency on the internet. In *Proceedings of first ACM Conference on Computer and Communication security* (1993) 102-196.
13. Rivest, R. L. and Shamir, A. PayWord and MicroMint: Two simple micropayment schemes. *CryptoBytes*, vol. 2, num. 1 (1996) 7-11
14. Sirbu M. and Tygar, J.D. NetBill: An electronic commerce system optimized for network delivered information and services. In *Proceedings of IEEE Compcon* (1995) 20-25
15. Sullivan, B.: 'Tis the season for credit-card heists, *ZDNet*, <http://www.zdnet.com/zdnn/stories/news/0,4586,2668427,00.html> (2000)
16. Treese, G. W. and Stewart, C.: *Designing systems for Internet commerce*. Amsterdam: Addison Wesley (1998)
17. Wayner, P. *Digital cash: Commerce on the net*, 2nd ed. London: AP Professional (1997)

An E-check Framework for Electronic Payment Systems in the Web Based Environment

A.R. Dani and P. Radha Krishna

Institute for Development and Research in Banking Technology (IDRBT), Castle Hills,
Masab Tank, Hyderabad – 500 057. Andhra Pradesh, India.

{ardani, prkrishna}@idrbt.ac.in

Abstract. The explosion of Internet and World Wide Web is rapidly changing the way the business transactions are carried out and it is emerging as a medium through which the goods and services are being provided to the customers. One of the main bottlenecks in the growth of E-Commerce is lack of suitable Payment Instrument and corresponding Electronic Payment System. To enable the E-Commerce to gain wide acceptance the secure electronic payment capabilities must be built up into the system. In real commercial transactions the payments can be token or notational. There are some attempts to build up both types of payments in E-Commerce transactions. In the paper, we have presented the architecture for E-Check and a trust model for secure transactions to support Electronic Payment System over Web. We have also discussed the protocols for the implementation of E-Check. Finally, the framework to support Electronic Payment Systems and related issues are presented. The proposed framework ensures the secure payment capabilities using E-Check that can be built up in E-Commerce environment in more effective manner.

1. Introduction

The explosion of Internet and World Wide Web is rapidly changing the way the business transactions are carried out. It is emerging as a medium through which the goods and services are being provided to the customers. This medium is being used to serve the customers at faster rate, and at lower cost. It is also being used to provide the better quality of service. Most business transactions involve the payment and settlement process. Electronic payment systems have become necessary to enable the Internet commerce. The requirements of payment transactions include confidentiality, security, integrity, non-repudiation and authentication. One of the main bottlenecks in the growth of E-commerce is lack of suitable payment instrument and corresponding electronic payment system.

Security is major concern in the payment systems. The systems without secure payment capabilities are similar to electronic advertising and ordering systems. Thus, to enable the e-commerce to gain wide acceptance, the secure electronic payment capabilities must be built into the system. The Secure Electronic Transfer (SET) protocol, which is used in credit cards, is gaining more acceptability [20] in providing security for electronic payment.

There have been several attempts to develop electronic payment systems [4,5,7]. In real world, the commercial transactions use different variety of payment instruments such as coins, paper cash that represents token payments; checks, drafts etc. that represent notational payments. The widely accepted payment methods for electronic commerce that are currently in use are credit card and Electronic Fund Transfer

(EFT). Different methods for payments are required for different purposes under different conditions. For example, credit cards are popularly used for customer retail transactions. However, they are not suitable for high-valued transactions, which is major requirement in business-to-business e-commerce.

The electronic check or E-Check is a dematerialized form of a paper check. It offers certain significant advantages over other electronic instruments for e-commerce transactions. For instance, as in paper check, the E-Check is useful for high-valued payments, to issue postdated checks, etc. A check is an order to the bank to pay the specified amount of money from the account of payer to the person named (payee) therein on or after a specified date. The payers and payees can be individuals, companies, government, banks, financial institutions, etc. In case the payer and payee have accounts in different banks, the check must be processed through a central clearing system. Nevertheless, paper checks are most widely used payment instruments all over the world after cash.

In the present work, we consider the electronic payment systems in which the payment instrument is E-Check. We present the architecture for E-Check and a trust model for secure transactions to support electronic payment system over web. The E-Check was designed based on the Financial Services Technology Consortium (FSTC) model [14]. It employs the digital signature, digital certificate and public key cryptography to deal with the security related issues. This paper also describes the necessary protocols for E-Check implementation. The proposed framework allows default-risk-free for payees.

The rest of the paper is organized as follows. In section 2, we present the related work and discussed the basic concepts of E-Check. In section 3, we provide the design architecture for E-Check. The protocols and implementation issues, specific to the Indian Banking industry, are discussed in section 4. Finally we conclude in section 5.

2. Overview of Electronic Payment Systems

Currently five electronic payment systems are available on the Internet. They are electronic cash, electronic fund transfer, credit cards, debit cards and electronic checks. The main differences between these systems pertain to anonymity of payer and payee, level of default risk to payee, permission of credit to payer and the level of authorization.

Electronic cash : The electronic cash systems replicate the cash over Internet. An efficient electronic cash system should possess all the properties of paper cash such as anonymity, intractability, transferability and fungibility. Electronic Cash [3], Digicash [12], Cybercoin [10] and Netcash [8,17] systems represent the electronic cash systems that are used in E-commerce transactions. These systems try to match the properties of cash like anonymity and intractability as closely as possible. For example, the Digicash system provides double spending detection. If the electronic cash is spent once all the parties remain anonymous. The identity can be traced in case of double spending. These systems do not possess the property of transferability. A transferable E-cash payment system, similar to paper cash, is proposed in [2]. Millicent and Netbill [17] proposed a system for micro payments that handle the transactions of small value. The electronic cash is still not widely acceptable and many legal issues are still being discussed [11,17,12].

Credit cards : The payment using Credit card is the most popular form of electronic payments for cyber shopping [19]. Most of the cyber shopping sites employ Secure Socket Layer (SSL) Protocol for security and encryption. Visa and MasterCard have proposed the SET protocol for secure electronic transactions using credit card. Credit card is equipped with public key encryption as well as certificates and digital signatures. It requires online authentication and payment gateway. Due to this the cost of transaction using credit card is comparatively higher. The credit card is risky instrument for the banks as well as the card holders. However, the merchant is assured of the payment once the payment transaction is authenticated by the system.

Debit cards : Debit card usage for payment become popular in some countries. These are similar to credit cards in operation from the merchant's point of view, but differs in credit extending from card holder and bank point of view. The main advantage of debit cards is the security for the issuing bank, because the holder has to pay the amount in advance. The major concern is who can issue such cards and whether it should constitute the demand and time liability [10,11, 12].

Electronic Fund Transfer (EFT) : EFT originally started on non-internet network to transfer funds electronically. It is gaining much popularity and is being attempted on the Internet. The Quickpay System of Security First Network Bank (SFNB) can be classified as an Internet-based electronic fund transfer service [21].

Electronic Check: The electronic check (or E-Check) resembles the ordinary paper check. The concept of electronic check is initiated by the Financial Services Technology Consortium (FSTC) and introduced several basic characteristics for E-Check [13][14]. It employs encryption, certification and tamper proof smart cards to deal with the security issue.

Figure 1 illustrates the standard functional-flow scenarios for E-Checks proposed by FSTC [14]. These are (a) Deposit and Clear (b) Cash and Transfer, (c) Lockbox and (d) Fund Transfer. The first two include a clearing process and are not completely safe from the risk of default. The last two are actually variant form of EFT. However, the design and implementation of a specific E-Check system is mainly based on the network connectivity between the entities (such as Payer, Payee, Payer's bank and Payee's bank) involved. The E-Check systems proposed in the literature are NetCheque [7,9], VirtualPin [15], NetBill [16], PayNow [10], NetChex [7,18] and SafeCheck [6]. However, none of the systems give sufficient attention to reduce the risk due to the defaulters and provide trust between the banks and users. In the present work, these issues are discussed in the design and implementation of E-Check system considering the FSTC model.

This paper describes an E-Check system that addresses the problems such as risk of default for the payee, and trust between the banks and users. Figure 2. shows the proposed E-Check model, which is a specific case of Figure 1(a). Though the E-Check model resembles the paper check system, it mainly facilitates the speedy process. The E-Check system consists of the various stages that will be carried out electronically: (a) The E-Check writer (payer) writes the E-Check using an electronic device like personal computer and sends the signed E-Check to the payee, (b) The payee deposits the E-Check with his bank. The payee's bank sends the E-Check for clearing to the paying bank, (c) The paying bank validates the E-Check. On acceptance, it debits the account of the E-Check writer, and (d) Once the E-Check is paid, it informs the payee's bank and the account of the payee is credited.

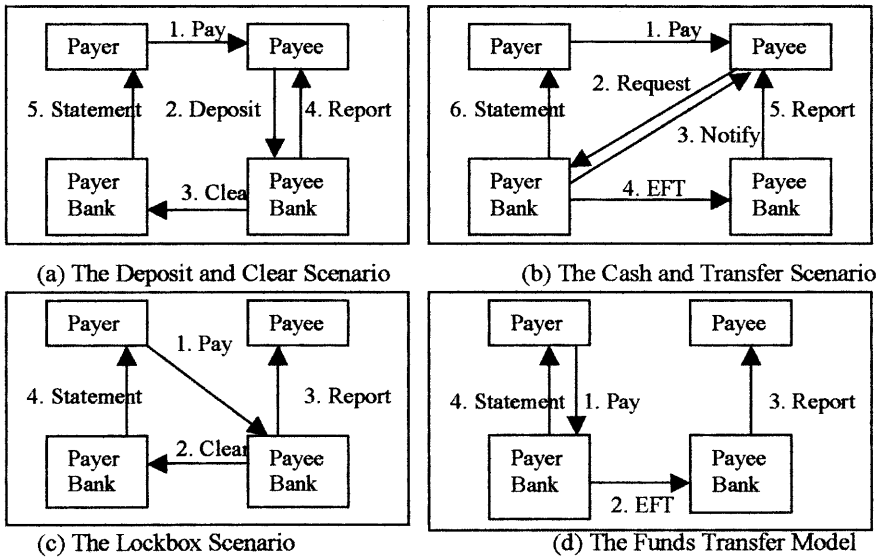


Fig. 1. Functional Flows Scenarios for Electronic Checks [14]

E-Check can be an effective payment instrument over Internet. It offers the speed, cost-effectiveness and reliability without requiring the large-scale investment and restructuring of existing systems and processes of the banks/businesses. An E-Check system can be made offline, and can be used without bilateral agreements or real-time link-ups. The major advantage of E-Check over Debit/Credit cards is that it can facilitate the high-value transactions, business-to-business payments and easy integration with existing systems.

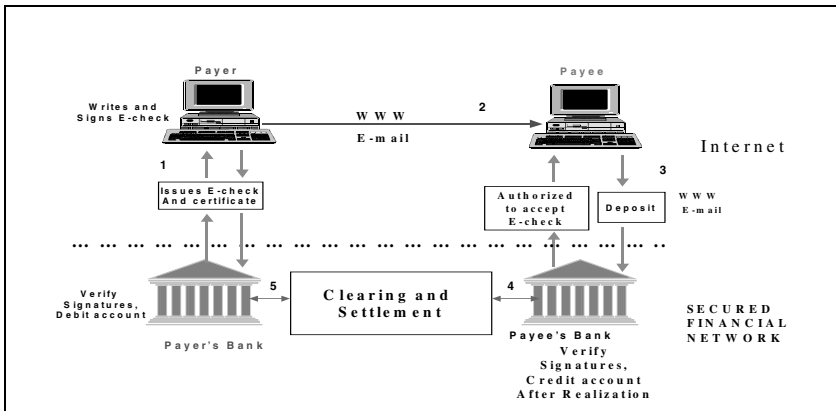


Fig. 2. E-check Model

3. Proposed E-check Architecture

The entities involved in E-Check system are payer, payee, payer bank, payee bank and clearing-house. The proposed E-Check system architecture is depicted in Figure 3. This E-Check system is a new type of electronic payment system that classifies the check writer (signer) and prevents (or at least reduces) the default risk on the payee's side. The system is basically similar to "Deposit and Clear" model in FSTC classification. Unlike FSTC model, the presented system is defined for centralized customer accounts in a bank.

According to FSTC model, the E-Check is an electronic financial document consisting of different blocks. In bank's view an E-Check in its life stage goes in four stages: Issue, Writing, Deposit and Payment. The structures of different blocks designed in the present work are as follows:

(i) E-Check Issue

{<Issue-and-Payer-Account-Information>, <Payer-Certificate>, <Bank-Certificate>, <Bank-Signature>, <Check-Information>}

(ii) E-Check Writing

{<Issue-and-Payer-Account-Information>, <Payer Certificate>, <Bank Certificate>, <Bank Signature>, <Check-Information>, <Action>, <Payer Signature>}

(iii) E-Check Deposit

{<Issue-and-Payer-Account-Information>, <Payer-Certificate>, <Bank Certificate>, <Bank Signature>, <Check-Information>, <Action>, <Payer Signature>, <Deposit Account Information>, <Depositor's Certificate>, <Depositor Signature>, <Bank Certificate>, <Bank Signature>}

(iv) E-Check Payment and Archive

{<Issue-and-Payer-Account-Information>, <Payer-Certificate>, <Bank Certificate>, <Bank Signature>, <Check-Information>, <Action>, <Payer Signature>, <Deposit Account Information>, <Depositor's Certificate>, <Depositor Signature>, <Bank Certificate>, <Bank Signature>, <Payment Data>, <Archive> }

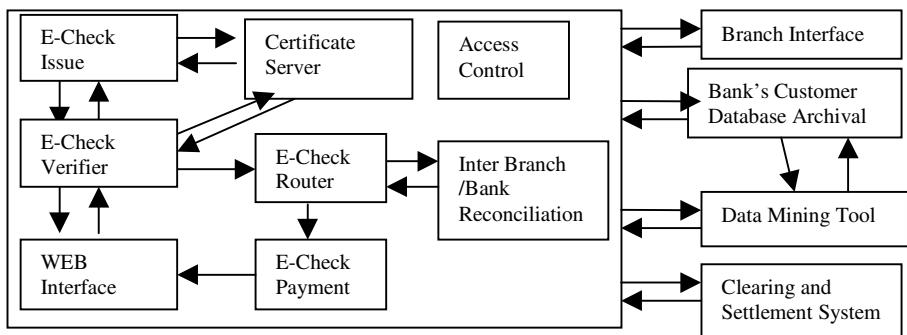


Fig. 3. E-check System Architecture

The <Check-Information> block contains the check specific information such as check ID (unique), Payer's Bank code, Payer's Account Number, Mode of operation (i.e., number of signatures required), Date and Amount. The <Payment Data> block, which contains the payment specific information, gets added with E-Check when it is paid. Lastly, the E-Check transaction information is archived for future purposes.

The trust model for E-Check system is designed as follows. In this system, Payer, Payee and Banks have digital certificates and key pair (public key and private key). The banks acts as Certification Agency for E-Check purpose and issue the digital certificate to its customer in X.509 format. The customer (payer) signs the E-Check using the digital certificate, which was issued against a particular account. If an account has more than one holder and its mode of operation is “Joint”, then a certificate is issued to each holder, and all holders must sign the E-Check. If the mode of operation is “Anyone” or “Survivor”, E-Check can be signed by any one of the holder. At the time of issuing the E-Check, the account related information is encrypted by the bank with a key which is only with the bank. The other information is digitally signed with payer’s private key, and then, optionally, encrypted using public key of the payee. The E-Check also contains the bank’s digital certificate. When the payee deposits the E-Check in his bank, he can encrypt it with the public key of the bank. The advantage of this model is that the incorrect payee can be traced using the digital certificate. The blocks for the trust model are :

{ (<Bank-Cert> Bank Certificate), (<Bank-PK> Bank’s Public Key),
 (<Payer-Cert> Payer’s Certificate), (<Payer-PK> Payer Public Key),
 (<Payee-Cert> Payee Certificate), (<Payee-PK> Payee Public Key) }

The E-Check book issued by the bank can be loaded into the floppy or smart card. A PIN is also provided by the bank to unlock the floppy/smart card, which can be changed by the customer later. At the time of writing the E-Check, the customer has to insert the floppy/smart card and unlock it using the PIN. This makes the system more reliable because the smart card, which has E-Check book, stores the keys (private and public key pair for the signature) using which the signed E-Check can be created offline. It not only provides adequate security (without any additional on-line authentication), but also takes care of loss of E-Checks. Thus, by storing the E-Check book on a floppy or smart card, one can most safely deploy the system. However, in case the smart card/floppy is lost, the payer can give a stop payment request to the bank for the remaining checks and a request can be made for new E-Check book. In our present prototype system, the E-Check book is not issued on-line due to the poor connectivity among bank and its branches (see section 4).

A data mining tool is developed to classify the customer segments using the association and classification techniques. The input to this system is E-Check database consisting of E-Check transactions detail, which was archived during the E-Check payment process, and customer demographic details. The information obtained from the association rules is used in selecting the relevant parameters to classify the data. The integration of this mining tool with the E-Check system prevents the possibility of bad defaults. It allows banks to minimize the risk of default on the payee’s side.

4. Implementation Issues

This section discusses the implementation of E-Check as a specific case of Indian banking system. Currently, there are several banks and their branches are not yet fully computerized, and also the connectivity among them is very poor. Hence, the present system is implemented as a centralized system assuming that the payer and payee have their account in the same bank. Four protocols are constructed for the

development of E-Check system (Fig. 4): (a) E-Check Issue Protocol, (b) E-Check Write protocol, (c) E-Check Deposit Protocol and (d) E-Check Payment Protocol.

(a) **E-Check Issue Protocol:** Customer, say A, of a bank, say B, initiate the E-Check Book issue process. The following messages and data are exchanged for the issue of E-Check between the customer, bank system and E-Check system:

- (i) A requests the bank B, through web browser or e-mail, by providing his account Details and confidential PIN for the issue of E-Check book.
- (ii) The E-Check system validates the details provided by A and verifies with its database and submit to the data mining system for A's loyalty to B. If the verification process is successful, then A's request is accepted and requested to submit his public key (PubKey) to the system. Otherwise, the request of A is rejected.
- (iii) The customer A submits his public key(Pubkey(A)) to the E-Check system and retains the private key.
- (iv) The E-Check system accepts the public key (PKA) and gets x.509 certificate (BC) for the Certificate server. It prepares the E-Check book and sends it back to the customer A.

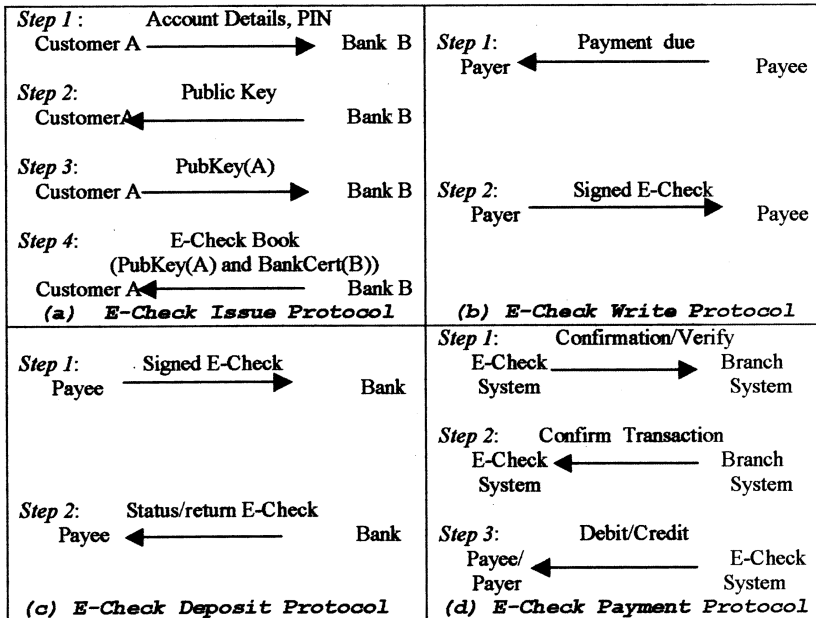


Fig. 4. E-check Protocols

(b) **E-Check Write Protocol:** At the time of E-Check writing the following messages are exchanged between the payer and the payee.

- (i) The payee sends the payment due information to the payer. (Optional).
- (ii) The payer writes the E-Check, signs it with his private key and sends it to the payee.

(c) **E-Check Deposit Protocol:** The payee deposits the E-Check in his bank. Two steps are involved in E-Check Deposit Protocol.

- (i) Payee deposits the signed E-Check in his bank.
- (ii) The bank accepts the E-Check and informs the depositor about its status or returns the E-Check.
- (d) **E-Check Payment Protocol:** Once the E-Check is deposited with the banks the flow of actions is as follows:
 - (i) The bank verifies the certificates and signatures of the received E-Check. The bank also validates the Issue and Account Information, Check Information and Payee details and generates the balance confirmation request to branches having payer's account.
 - (ii) The branch confirms the balance. In the case of low balance, the check is rejected and a notification is sent to the payee. This information is also recorded in the database, which will be useful for mining.
 - (iii) The account of payer is debited and that of payee is credited.

The payer and depositor account information is encrypted with a key, which is with the bank. At the time of writing E-Check, the payer digitally signs E-Check using his private key that can verify by the certificate issued to him by the bank. In the same way, depositor also signs E-Check using his private key. If required the payer can encrypt the E-Check using the public key of payee, and payee can encrypt using public key of the bank so that the vital information is encrypted. Secure Socket Layer protocol can be used for additional security during the E-Check issue process. The various modules in the implemented E-Check system are:

(a) *E-Check Issue* – processes E-Check Issue, Renewal, Certificate renewal and Revocation, and Stop Payment requests; (b) *Certificate Server* – issues the digital certificate in X.509 format to the bank's customers and performs the certificate and key management functions; (c) *E-Check Writing* – handles the E-Check writing and deposit operations of the customers; (d) *E-Check Verifier* – verifies the E-Check received from the payee and validates the certificates, signatures and syntax of various blocks; (e) *E-Check Router* – separates the E-Check bank wise and sends them to the external systems like Clearing and Settlement System, and also to the different branches of the bank; (f) *E-Check Payment* – handles payment request and avoids the duplicate payments; (g) *Inter Branch Reconciliation* – reconciles the debit requests, confirmations and credit requests of different branches of same bank and other banks; (h) *Branch Interface* – provides the interface between the bank's operational system and E-Check System; (i) *WEB Interface* – provides the WEB interface for the customer which includes the facilities like writing, depositing E-Check, sending Stop Request, Query Status, Request for Renewals etc; (j) *E-Check Archival System* – archives all paid E-Checks and keys and handles back up and restore functions; and (k) *Access Control System* - controls and manages the access roles of users to E-Check System.

The *data mining system*, which is integrated with E-Check system, helps in avoiding the default risk. The Apriori algorithm [1] is used for association rule and neural network for classification. Based on the customers' information and E-Check information, the system produces the hidden relationships in the data and predicts the level of customer loyalty to the bank at the time of issuing or renewing the E-Check. The E-Check system is developed using JAVA 1.2 and JDBC and the back-end database is in ORACLE. The system is general and can be used with any RDBMS, and support Windows and Linux operating systems. One nationalized bank in India is considered for pilot implementation of this system.

5. Conclusion

Internet, WWW and Electronic Commerce are becoming important channels for current e-business. In this paper, we presented a frame work for electronic payment systems using E-Check. We considered the case where a bank can have centralized customer accounts and E-Check book is issued from one point in the bank. The necessary protocols and the trust model for E-Check are discussed. Integration of Data Mining techniques with E-Check system allows a substantial reduction of bad defaults. The proposed framework ensures the secure payment capabilities using E-Check that can be built up in E-Commerce environment in more effective manner.

Acknowledgements. The present work is part of “Technology for E-Commerce” project funded by Ministry of IT, Government of India. The authors thank V.P. Gulati, Director, IDRBT and the project team members for their support during this work.

References

1. R. Agrawal, T. Imielinski, and A. Swami, Mining association rules between sets of items in large databases, Proceeding of the ACM SIGMOD International conference on Management of Data, Washington, USA, May 1993.
2. R. Anand Sai, C. E. Veni Madhavan, An Online Transferable E-Cash Payment System, INDOCRYPT 2000, LNCS
3. D. Chaum, A. Fiat, M. Naor, Untraceable Electronic Cash. Advances in Cryptography – CRYPTO’88, LNCS, Volume 403, 1990
4. A. Crede, Electronic commerce and banking industry. The requirement and opportunities for new payment systems using Internet. (jcmc.huji.ac.il/vol1/issue3/crede.html).
5. R. Kalakota, and A. B. Whinston, Frontiers of Electronic Commerce, reading MA: Addison-Wesley, 1996
6. J. K. Lee and H. S. Yoon, An Intelligent Agents-Based virtually defaultless check system: The SafeCheck system, International journal of Electronic Commerce, Vol. 4, No. 3, pp. 87-106, 2000.
7. B.C. Neuman, and G. Medvinsky, Requirement for network payment: The Netcheque™ perspective, Proceedings of IEEE Compcon ‘95, San Francisco, March 1995.
8. B.C. Neuman, and G. Medvinsky, B.C. NetCash A design for practical electronic currency on the Internet. In proceedings of First ACM Conference on Computer and Communications Security, 1993 (www.isi.edu).
9. B.C. Neuman, and G. Medvinsky, Requirement for network payment: The Netcheque™ perspective, In proceedings of IEEE Compcon’ 95, San Fransisco, March 1995.
10. www.cybercash.com/cybercash/paynow, CyberCash, Inc. PayNow™ pilot programme 1997.
11. CyberCash, Inc., CyberCash –Information, www.cybercash.com
12. www.digicash.com/products/project, DigiCash, DigiCash products – the CAFÉ Project
13. www.echk.org
14. Financial Services Technology Consortium. E-Check project, www.fstc.org
15. www.fv.com/demo, First Virtual Holding. The First Virtual Solution, 1997
16. The NetBill Project. 1997 (www.ini.cmu.edu/netbill)
17. C.M.U.’s I.N.I. Project. The NetBill Project 1997, www.netbill.com,
18. www.netchex.com, NetIInc. NetChex Homepage, 1996

19. Netscape,SSL Specification – home.netscape.com/eng/ssl3/index.html
20. MasterCard/Visa, Secure Electronic Transaction (SET) Specification, (www.setco.org/set_specifications.html)
21. Security First Network Bank, Bank Demonstration, 1997 (www.sfnb.com/demos/bankdemos.html)

Trader-Supported Information Markets - A Simulation Study

Michael Christoffel¹, Thorwald Franke², and Stefan Kotkamp³

¹ Fakultät für Informatik, Universität Karlsruhe, Postfach 6980
D-76128 Karlsruhe, Germany
christof@ira.uka.de

² Deutsche Börse Systems AG
D-60485 Frankfurt am Main, Germany
thorwald_franke@exchange.de

³ Fakultät für Wirtschaftswissenschaften, Universität Karlsruhe, Postfach 6980
D-76128 Karlsruhe, Germany
stefan.kotkamp@wiwi.uni-karlsruhe.de

Abstract. The modern society depends on the provision and distribution of information. We observe the development of world-wide information markets. Traders play an important role in these markets, as they bring together supply and demand. In this paper, we describe a simulation study about mechanisms and rules in information markets under special consideration of the role of traders. The usefulness of simulation for market analysis is shown in selected experiments.

1 Introduction

The modern society depends on the provision and distribution of information. The work of scientists, economists, politicians, educators, and others is affected by the steady supply with up-to-date information. Hence, we observe the evolution of word-wide information markets, where the value of a piece of information is determined by the law of supply and demand. We can expect that electronic trade will become dominant in these modern markets [1]. Since information goods can not only be digitized easily, but also edited and distributed electronically via the Internet, they are particularly well-suited candidates for electronic publication and commerce.

Information markets have some special features that distinguish them from traditional markets: the unequal distribution of cost, the possibility of cheap and maybe unauthorized copies and manipulations, the central role of quality of service, and the importance of services in addition to the pure delivery. A good general introduction into information markets can be found in [2], a more theoretical approach in [3]. An overview on pricing strategies for information goods is contained in [4] and [5].

In traditional markets intermediaries play an important role. There are indications that this will remain true for electronic markets [6]. Intermediaries are market participants other than customers and providers, which facilitate, aggregate, mediate, ensure trust, or provide market information. Information intermediaries are necessary to remedy one of the greatest problems that arise from the new richness of information

providers, namely information overload. Customers are often not able to find and to compare suitable providers, and providers can not reach their customers by marketing. One such class are traders.

A simple and general definition of a trader can be found in [7]: “A trader is a third party object that enables the linking of clients and servers in a distributed system.” Transferred to the market scenario, we can define: A trader is a market object that brings together supply and demand. [8] gives a general overview on trading in information markets and communities of agents.

In this paper, we present a study on the rules and mechanisms in an open, dynamic, heterogeneous, and distributed information market, using the method of simulation. Our focus of interest hereby is the effect of traders on this market. It is important that traders are market participants without special privileges. The scenario of an open market must be distinguished from auctions and stock markets with a monopoly position of the brokering service. We hope to show that simulation is more appropriate for still developing markets than a closed mathematical model or an empirical study.

The paper is organized as follows: In section 2 we introduce an earlier approach to a simulation-based analysis of trader-supported information markets. In section 3 we present our simulation model and explain our experiments. The major results of these experiments are summarized in section 4. In the subsequent sections we discuss some of the experiments in more detail: Section 5 contains experiments concerning basic market mechanisms, and section 6 contains experiments concerning the influence of traders. We conclude the paper with section 7.

2 Related Work

A theoretical approach to the analysis of traders and intermediaries in information markets can be found in [9]. An interesting simulative approach is by Kephart and Hanson from IBM Thomas J. Watson Research Center [10,11,12].

Kephart and Hanson modeled an information market using three agent types: sources, brokers, and consumers. Sources produce articles and offer them to brokers. Brokers purchase these articles and try to resell them to consumers. The decisions of consumers and brokers are based on price and utility functions. Kephart and Hanson also considered the market infrastructure and took into account communication and delivery costs. The focus of their work was on the profit maximization and strategies for the brokers.

The approach makes a number of unusual assumptions. Commerce transactions are triggered by the providers, and the role of the customers is limited to the decision whether to buy or not. However, in electronic trade commerce transactions are usually triggered by customers that declare their demand. Brokers act as middlemen, buying and reselling goods physically. This ignores a major advantage of electronic commerce, namely that there is no need for a delivery before the trade is complete. So intermediaries better act as mediators and facilitators, and not as middlemen.

Despite these inconsistencies and the preferential treatment of the intermediaries, it is an interesting approach, bringing some pioneering results:

- Competition between brokers leads to (cyclic) price wars. These price wars were harmful to both brokers and customers. In many cases, brokers were eliminated.
- Some brokers avoided competition and price wars by specializing and niche-finding.
- Specialization could also be observed without price wars. Brokers were able to increase their profit by specialization.

3 Simulation Model

In this section, we describe our own simulation model KASTIM (KARlsruhe Simulation of a Trader-supported Information Market). The model underlies a three-tiered structure of customers, providers, and traders, which are all part of the simulation. The task of the traders in this model is the mediation between customers and providers. However, customers and providers may also interact directly.

The simulation model supports many features that allow to estimate the developing information markets as close as possible:

- clear distinction between infrastructure, search, and product costs
- different charging models applicable between market participants
- influence of advertisement
- cooperation among traders
- quality of service attributes that can be used to describe the confidence with delivery ways and times, formats, ease of use, response times, etc.
- memory function and learning behavior of customers, providers, and traders
- migration and immigration of market participants

Of course, there are also simplifications in comparison with real markets. For instance, we do model product groups rather than single information products. Product availability is a measure for the supply of the providers in a product group.

While the benefit of traders and providers is equated with their profit, the benefit of a customer depends on several factors. The aggregation of the factors is expressed by a utility function:

$$\text{utility} = \begin{cases} (1 - \text{costPriority})\text{servFactor} + \text{costPriority} \cdot \text{costFactor}, & \text{if success} \\ \text{failFactor}, & \text{otherwise} \end{cases} \quad (1)$$

$$\text{servFactor} = 1 - \sqrt{\frac{\sum_{i=1}^{\text{numSAttr}} (\exp \text{SAttr}_i - \text{realSAttr}_i)^2}{\text{numSAttr}}} \quad (2)$$

$$\text{costFactor} = 1 - \frac{\text{realSearchC} + \text{realProdC} + (1 - \text{adAggr})\text{realTradC} + \text{realNetC}}{\text{maxSearchC} + \text{maxProdC} + \text{maxTradC} + \text{maxNetC}} \quad (3)$$

$$\text{failFactor} = \frac{\text{realSearchC} + \text{realTradC} + \text{realNetC}}{\text{maxSearchC} + \text{maxTradC} + \text{maxNetC}} \quad (4)$$

In the above equations, costPriority denotes the priority of the customer for costs (i.e., low prices) as opposed to the quality of service (range $[0,1]$). numSAttr is the number of service attributes, expSAttr_i and realSAttr_i are the expected and the real value for the service attribute i (range $[0,1]$). Costs are composed from search cost, product cost (prices), trader cost, and network cost (infrastructure). E.g., realSearchC and maxSearchC are the real and maximal search cost. adAggr is the advertisement aggressiveness of a trader.

The complete algorithm contains several more parameters. For example, if the advertisement aggressiveness exceeds a limit individual for a customer, the customer feels annoyed, and the trader will be ignored.

In our model, we apply event-based simulation [13]. Each searching and buying action is triggered by the customer. Whether the customer buys a product or not depends on the offers and the cost. However, incurred search costs have always to be paid. When a customer develops the demand for a special product, he/she starts a search for this product. Either he/she asks known providers, or he/she contacts a trader. The relations between customers, providers, and traders are not fixed and can be changed at any time.

The experiences of the market participants plays an important role, too. All participants hold data about other participants. For example, the decision whether a customer uses the services of a trader or not and which trader he/she selects depends on the utility the customer received from the trader in previous queries (but there is also a curiosity factor that induces the customer to test new traders or providers).

In order to enable the market participants to optimize their strategies, we have introduced a 'trial and error' behavior. Participants make slight changes to strategic variables at random and observe the change of the values that are to be optimized. For example, a provider may vary stocks and prices to optimize its profit.

The simulation model has been implemented under Java 2. The experiments were executed on a Unix workstation with a 296MHz 13,1 SPECint processor and 128 MB RAM. A simulation run with 4 customers, 20 providers, and 4 traders over 1300000 units of time took about 7 minutes of computing time.

4 Summary of Results

We have carried out 16 different experiments with the KASTIM system until now, each encompassing several individual simulation runs. We come to the following major results:

- General principles of information markets as forecast by the theory or already shown in other experiments, such as price wars, niche-finding, and the continuous dynamism of the market, can be confirmed.
- Competition and price wars between providers can be harmful for the customer if this leads to a reduction of the product availability.
- Competition and price wars between traders have less influence on the customers, because disadvantages by fighting traders can be compensated.
- The installation of trading services in an information market leads to a clear and permanent increase of the utility to the customers.

- The specialization of providers decreases the profit of the traders, since customers do not have to rely on the help of the traders if they already know a specialist for their special demand.
- Traders profit from market dynamism, because their market knowledge becomes more valuable.
- Traders profit from a steady renewal of the customer clientele, because customers participating in the market for a longer time learn from previous queries about the market and decreasingly need the help of trader.
- Unsystematic cooperation between traders has no effect.
- The needs of customers that give priority to cost and those that give priority to services are completely different. Traders can benefit from this knowledge by specialization.
- For a provider, it is sufficient to register with only one trader. However, providers can profit from registering at more than one trader if the customers give priority to services rather than to cost.

In the following sections, we go into more detail on some selected experiments.

5 Basic Market Mechanisms

In this section, we describe two experiments about two basic market mechanisms: competition and specialization among providers.

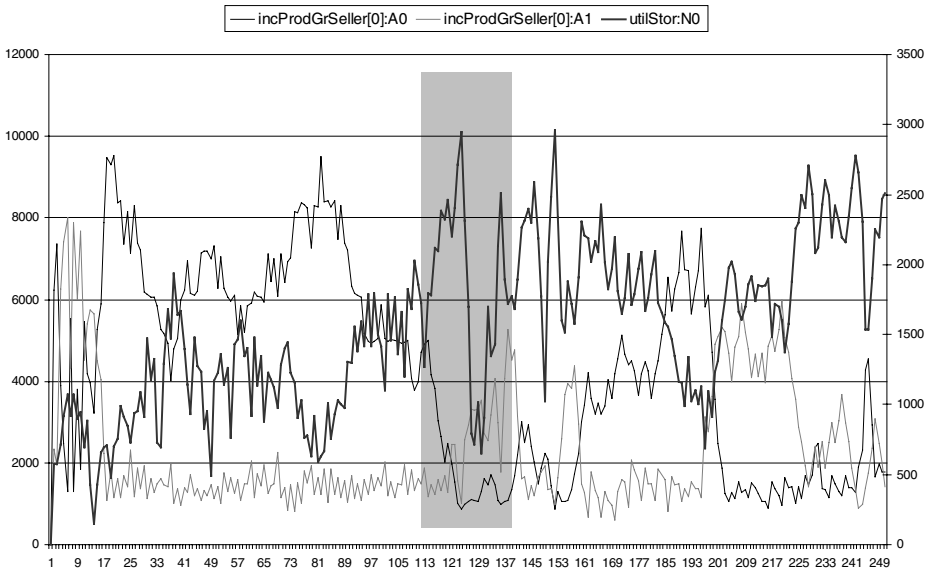


Fig. 1. An experiment to investigate competition between two providers. The figure shows the development of the income of the two providers (`incProdGrSeller[0]:A0` and `incProdGrSeller[0]:A1`) and the utility of the customer (`utilStor:N0`) over the time

In order to investigate competition, we simulated a market with only 1 customer active at any time, 2 providers and no traders over 300000 units of time. Both providers are set up identically, and they offer the same single product group. From the customer's point of view, they only differ in their search and product costs

Figure 1 shows the development of the income of the two providers. The dynamism of the curves is caused by the steady movement of prices (and, as a consequence, sales). Measurements confirm that there are price wars between the providers, as both try to undercut each other continuously in order to achieve the market leadership (and raise the prices afterwards). Price wars affect both product and search costs.

The utility to the customer is also displayed in Figure 1. Here we can observe an interesting phenomenon (best visible in the marked detail): Before market leadership changes, the utility raises since prices fall, but then the utility goes down rapidly, although prices are low. An explanation can be found when we observe the development of the number of failed queries. The price wars led to an increase of the failed queries. This indicates that the customer was not able to find the demanded goods and could not benefit from the low prices. Indeed, the measurements show a reduction of the product availability of both providers, indicating a reduction of the stock. It is quite natural that the defeated provider reduces stock, because it sells less. The winning provider has also reasons to reduce stock, because its product prices are so low that it has to reduce stock in order to save storage cost. Another reason is that it sells so much that the falling product availability does not matter.

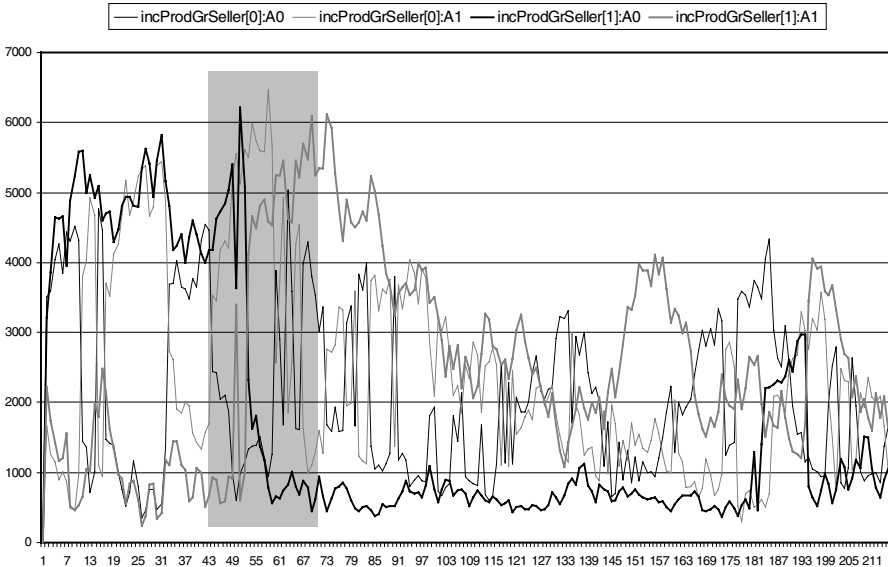


Fig. 2. An experiment to investigate specialization between two providers. The figure shows the development of the income of the two providers in product group 0 (incProdGrSeller[0]:A0 and incProdGrSeller[0]:A1) and in product group 1 (incProdGrSeller[1]:A0 and incProdGrSeller[1]:A1)

With the next experiment we illustrate the phenomenon of product specialization. As a significant difference to the previous experiment, there are now two product groups. The other parameters are 2 customers, 2 providers, no traders over 300000 units of time. Figure 2 shows the income of both providers realized in the two product groups. The most interesting detail is marked here, too. We can see that before time point 49 (49000 units of time), provider 0 has a clear leadership in product group 1, and provider 1 has a clear leadership in product group 0. So we can confirm the specialization of the providers.

However, market niches do not remain unchallenged. After measure point 49 provider 2 successfully conquers market leadership in product group 1. Provider 0 tries to switch to product group 0, but the attempt to achieve a dominant position does not succeed. Measurements show the effect of market niches in the product availability, too. The market leader in this product group has a product availability of up to 100%, whereas the other provider reduces stock under 50% (however, it does not give up the product group completely). Simulation runs with 16 providers (and 4 product groups) show an oligopoly of a few providers rather than the dominance of one provider.

6 The Role of Traders

So far we presented baseline experiments with no intermediaries. We now introduce a second set of experiments that include traders. We are focussing on the value of traders for the customer, the influence of the number of providers and of the total number of customers on the trader's profit.

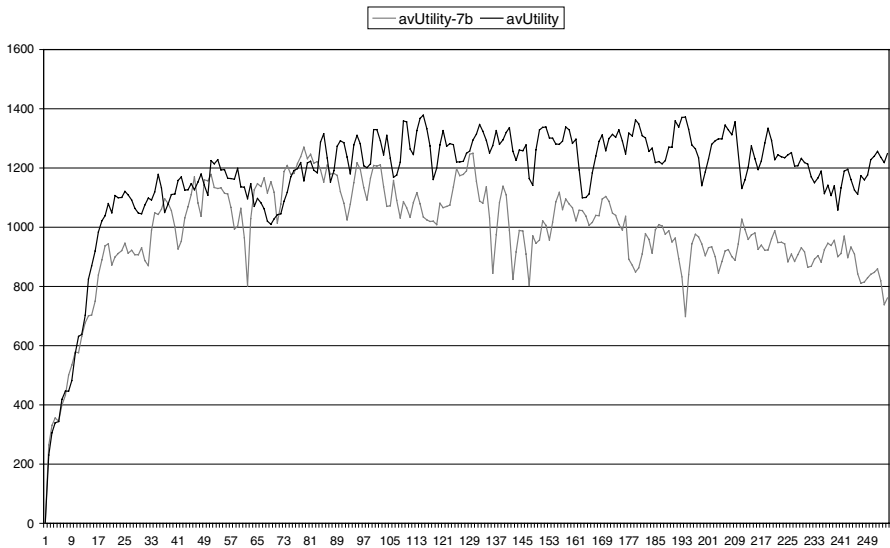


Fig. 3. An experiment to investigate the utility of traders to the customers. The figure shows the development of the average utility of the customers with traders (avUtility) and without traders (avUtility-7b)

The first question was whether customer benefit from traders or not. We compared the average utility to a customer in two markets: One with no traders, another one with 10 traders. The other parameters in both markets were 10 customers, 20 providers, and 5110000 units of time. As we can see from Figure 3, the average utility of a customer in the market with traders is approximately 19% higher than in the market without traders.

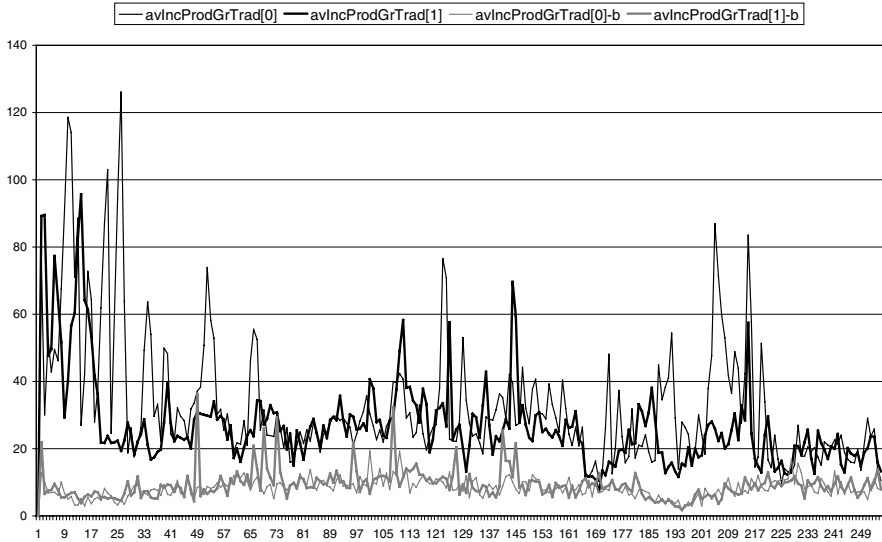


Fig. 4. An experiment to investigate the influence of the number of providers in a market on the profit of the trader. The figure shows the development of the average income of the traders in two product groups in a market with 4 providers (avIncProdGrTrad[0] and avIncProdGrTrad[1]) in a market with 24 providers (avIncProdGrTrad[0]-b and avIncProdGrTrad[1]-b)

In order to observe the influence of the number of providers on the trader profit, we simulated two markets, one with 4 providers, another one with 24 providers. The other parameters were 4 customers, 4 traders, and 1800000 units of time. In Figure 4 we see - on first glance surprising - that the trader's profit is higher in the smaller market. The explanation can be found in the specialization of the providers. Because of the high level of competition in the market with 20 providers, the providers concentrated on one or few product groups (niche-finding). So customers could find experts among the providers and needed the help of a trader less often.

In the next experiment we investigated the influence of the total number of customers on the trader profit. We simulated two markets: In the first market customers stay the same over the simulation run, and in the second market customers leave the market and are replaced by new customer (randomly, one customer every 500 units of time). The parameters in both simulations were 4 customers, 24 providers, and 4 traders over 1344000 units of time. Figure 5 shows the development of the average profit of the traders in both markets. The profit of the trader is approximately 37% higher in the market where customers are changing. So traders profit from a higher number of customers. The explanation lies in the memory of the

customers. During the simulation run, they gain their own knowledge about the market. However, the dynamism of the market prevents traders from becoming superfluous, even if the customers do not change.

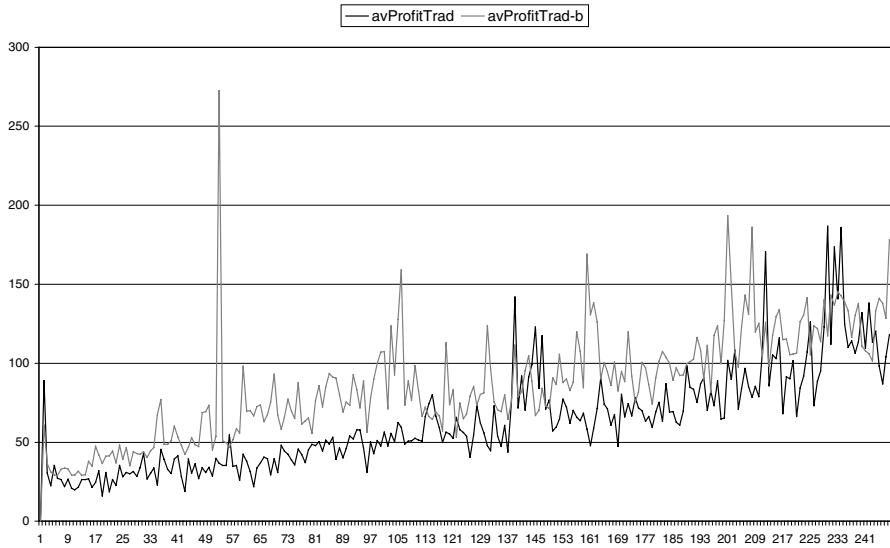


Fig. 5. An experiment to investigate the influence of the total number of customers in a market on the profit of the trader. The figure shows the development of the average profit of the traders in a market where customers are not replaced (avProfitTrad) and in a market where they are replaced (avProfitTrad-b)

7 Conclusion

In this paper, we presented a simulation study about the rules and mechanisms in an open information market under the special consideration of the role of traders. The aim of this study was to show how simulation can help in the analysis of market mechanisms, especially when these markets are in the phase of evolution. The understanding of the market mechanisms is essential when building the necessary information technology infrastructure. The consideration that market participants in new electronic markets will often be autonomous software agents rather than human beings even increases the importance of these questions.

The correspondence of our observations with the theory of the information market and the results of the study of Kephart and Hanson seems to confirm the correctness of our approach. This is not only true for obvious observations like dynamism and competition. Complex and somehow surprising phenomena could be observed, too. For example, the observation that price wars may harm customers has also been reported by Kephart and Hanson. We have found an explanation for this in the reduction of product availability. Basic strategies, such as differentiation and cost leadership, could also be confirmed. An important result of our study is that traders

are always worth their price if the market is sufficiently dynamic and heterogeneous. This result has been forecasted by Rose [9].

We plan to continue our work on information markets. We plan new, more thorough experiments, leading to a more precise description of the dependencies of the variables, and also longer simulation runs, showing the success of strategies on a longer time-scale. Extensions to the model are planned to include cooperations, negotiations, and trust-building measures. The implementation of an infrastructure for a special market [14] will produce a basis for experiments under realistic conditions.

References

1. Bakos, Y.: Towards Friction-Free Markets: The Emerging Role of Electronic Marketplaces on the Internet. In: Communications of the ACM, 41(8) (1998)
2. Shapiro, C., Varian, H.: Information Rules (1999)
3. Arunkundram, R., Bakos, Y.: Search Characteristics and Equilibrium in Internet-based Electronic Marketplaces. In: Proceedings of the 9th Workshop on Information Systems and Economics, Atlanta (1997)
4. Kotkamp, S.: Pricing Strategies for Information Products. In: Deutsche Bank - Inhouse Consulting Manager, 3 (2000)
5. Bakos, Y., Brynjolfsson, E.: Aggregation and Disaggregation of Information Goods: Implications for Bundling, Site Licensing and Micropayment Systems. In: Proceedings of Internet Publishing and Beyond: The Economics of Digital Information and Intellectual Property, Cambridge (1999)
6. Bailey, J., Bakos, Y.: An Exploratory Study of the Emerging Role of Electronic Intermediaries. In: International Journal of Electronic Commerce, 1(3) (1997)
7. Bearman, M.: ODP-Trader. In: Proceedings of the International Conference on Open Distributed Processing, Berlin (1993)
8. Klusch, M.: Agent-Mediated Trading: Intelligent Agents and E-Business. In: Journal on Data and Knowledge Engineering, Special Issue on Intelligent Information Integration, 36(3) (2001)
9. Rose, F.: The Economics, Concepts, and Design of Information Intermediaries (1999)
10. Kephart, J., Hanson, J., Levine, D., Grosz, B., Sairamesh, J., Segal, R., White, S.: Dynamics of an Information-Filtering Economy. In: Proceedings of the 2nd International Workshop on Cooperative Information Agents, Paris (1998)
11. Kephart, J., Hanson, J., Sairamesh, J.: Price War Dynamics in a Free-Market Economy of Software Agents. In: Proceedings of the 6th International Conference of Artificial Life, Los Angeles (1998)
12. Hanson, J., Kephart, J.: Spontaneous Specialization in a Free-Market Economy of Agents. In: Proceedings of the Workshop on Artificial Societies and Computational Markets, Minneapolis/St. Paul (1998)
13. Zeigler, B.: Theory of Modeling and Simulation (1985)
14. Christoffel, M., Pulkowski, S., Schmitt, B., Lockemann, P.: Electronic Markets: The Roadmap for University Libraries and their Members to survive in the Information Jungle. In: Sigmod Record Special 27(4) (1998)

An Integrated Framework of Business Models for Guiding Electronic Commerce Applications and Case Studies

Chien-Chih Yu

National ChengChi University
Taipei, Taiwan, 11623 ROC
ccyu@mis.nccu.edu.tw

Abstract. A well-designed business model serves as a key success factor for business enterprises to create electronic commerce (EC) applications and sustain profits. The lacking of uniform views and architecture for business models leads to the difficulties for supporting profitable EC-related business development and management, and also makes it uneasy to clearly outline the content and procedure for performing comprehensive EC case studies and comparative analyses. The goal of this paper is to propose an integrated framework of business models for efficiently and effectively guiding EC applications and case studies. Within this architectural framework a component structure of business models and a content structure of case studies which uses business models as the core are presented. Critical components such as assets, markets, customers, competitors, products, services, costs, prices, revenues, profits, market shares, economic scales, marketing strategies, competitive advantages and the dynamic relationships between these components are discussed.

1 Introduction

Rapid advancement of Internet technologies and fast growth of electronic commerce (EC) applications have greatly affected the ways companies doing businesses. The impacts of this evolving digital economy on business strategies and processes are so strong and eventually force almost all business enterprises to leverage their technology and information infrastructure for quick EC adoption to create better market and customer values and to attain competitiveness as well as profitability. One critical problem remained to be tackled for top management then is how to choose or build a suitable business model for successful EC implementation to gain and sustain competitive advantages, as well as to increase market share and finally make profits. Business model not only plays a determinant role in the survive-to-success campaigns of EC entrants but also serves as a foundation for professionals and academics to perform EC case studies as well as sophisticated comparative analyses. Business models emerge as extremely important management issues in recent years and have in fact become increasingly significant research topics for exploration. Although there are a few previous research works in the literature addressing business models, the definitions are inconsistent and the views for model formulation and classification

diverge from each other. No common architecture exists for uniformly specifying the component structure of business models and for illustrating the dynamic and interactive relationships between these components. As a result, topics related to business models are often considered as belonging to an area that is most discussed but least understood in the EC domain and thus deserve more efforts to be made on the development of related concepts, tools and applications. On the other hand, although EC case studies provide a good means for investigating and evaluating business performances, few works actually pin-point the central issues and capture the true spirits of business models from an integrative view. For handling the current shortfalls and fulfilling the need, the goal of this paper is to provide an integrated framework for EC business models to efficiently and effectively guide the processes of model formulation, specification and implementation, as well as of case studies. Two main portions contained in this architectural framework include a component structure of business models and a content structure of case studies in which the business model is the core of the case contents. By taking such an integrative view, critical components and elements such as markets, customers, competitors, products, services, resources, capabilities, assets, costs, prices, revenues, profits, economic scales, market shares, marketing strategies, and competitive advantages as well as the linkages and interactions between these components can be easily identified and thoroughly discussed. In addition, based on the proposed integrated framework and associated model and case structures, business categorization, performance analysis, process reengineering and strategic planning can be streamlined. In the mean time, specific EC business case studies and cross-business or cross-industry analyses can be systematically directed and executed to achieve the efficiency and effectiveness as expected.

2 Literature Reviews

Among a variety of inconsistent definitions provided in the literature, a business model is defined in the basic sense as a method of doing business by which a company can generate revenue [12]. Or it is defined as a method by which a firm builds and uses its resources to offer its customers better value than its competitors and to make money doing so, and can be conceptualized as a system that is made up of components including value, scope, revenue sources, price, connected activities, implementation, capabilities, and sustainability, as well as linkages and dynamics [1]. Or, it is an architecture for the product, service and information flows, including the description of the various business actors and their roles, a description of the potential benefits for the various actors, and a description of the sources of revenues [15]. Besides, types of business models in the literature are categorized in many different ways depending on views taken from different angles. Some examples of model classification are given below.

1. Depending upon the class of trading parties and the transaction environments, business models identified include Business-to-Business(B2B), Business-to-Consumer(B2C), Consumer-to-Consumer(C2C), and Intra-Business models [5,14,16,17].

2. Depending upon web site services and functions, generic forms of business models observed include Brokerage, Advertising, Infomediary, Merchant, Manufacturer, Affiliate, Community, Subscription, and Utility models. The Brokerage model can be further divided into B2B, B2C, and C2C markets [12].
3. Depending upon the types of electronic markets, eleven implemented business models selected include E-Shop, E-Procurement, E-Auction, E-Mall, Third Party Marketplace, Virtual Communities, Value Chain Service Provider, Value Chain Integrator, Collaboration Platforms, Information Brokers, and Trust Service Provider. Degree of Innovation and Integration of Functions are two dimensions for model classification [15].
4. Depending upon economic effectiveness and revenue sources, identified business models include Content, Advertiser, Intermediaries, and Relationships. The art of management is pointed out as the key success factor of EC business models [13].
5. Focusing on the B2B applications, three marketplace business models including buyer-oriented, supplier-oriented, and intermediary-oriented models are identified depending on the control roles of the marketplace. Other business models classified are virtual corporation, networking between headquarter and subsidiaries, and online services to business [16].
6. Concerning the contents of the Internet sales, business models such as selling goods and services, selling information or other digital content, and the advertising supported model, advertising-subscription mixed model as well as the fee-for-transaction model are classified [14].
7. Other business models related to specific application domains include those discussed in the cases of network publishing, marketplace for small and medium enterprises, networked direct sales, and electronic cash, etc [11,6,8,9].

The various approaches to describe and classify business models stated above have indeed indicated some directions to look into the model formulation issues and have also partially addressed some characteristics of the models and associated components. But the applied concepts and methods are inconsistent and do not fully encompass the various facets of business models. Therefore, an uniform conceptual architecture to integrate these diversified model construction and classification schemes as well as to provide a well-defined component structure for clearly illustrating entities and relationships of business models is still lacking and needs to be established. A desired integrated framework of business models should be able to take into account all aspects of model formulation, classification, implementation, and evaluation issues, to identify a set of critical factors for successful and profitable Internet-based businesses, as well as to guide the conduction of EC case studies. In the following sections, an integrated framework of business models to meet the need is presented. Within this framework, a component structure of business models with major components, elements, linkages, and interactions is provided first followed by the presentation of a content structure for case studies in which the business model is the core of the cases.

3 A Component Structure of Business Models

From an integrated view, the EC business model can be defined as a conceptual architecture for representing entities and relationships of model components with identified critical success factors of electronic businesses. EC business models can be used to develop executable methods for guiding companies to efficiently and effectively conduct businesses, gain competitive advantages, as well as sustain profits. Within the architectural framework, main components to be identified and described include market and supply chain participants, products and services, resources and capabilities, assets and costs structures, pricing and billing methods, revenues and profits sources, marketing strategies and competitive advantages, market shares and economic scales. Also described are associated elements and interactive relationships of these components. One of the reasons to build such a component structure for business models is for structurally representing and systematically measuring major influential factors of business performances. Among these factors, assets, costs, products, services, prices, channels, promotions, etc are associated with the internal market environment, and customers, competitors, supply chain partners are related to external market environment. Other market-related factors of the external environment that can be considered other than those already identified in the business models include stakeholders, technology impacts, international policies, as well as legal, political, and economical situations. Figure 1 illustrates the component structure of business models in which major components and interactive relationships between them are presented. Key components, elements, factors, and interactions are described below.

Markets:

Markets are trading environments for buyers and sellers which can be classified in several different ways including global, regional, national, or local markets by scope; business-oriented or consumer-oriented markets by targeted customers; catalog-based or auction-based markets by transaction functions; and books, cars, or medicine by product categories; etc. Markets can also be segmented by customer characteristics such as ages, incomes, and preferences.

Customers:

Customers are buyers of the markets with various types including individuals, businesses, and communities.

Products:

Products, physical or digital in form, belong to one kind of target objects of business transactions offered by sellers to potential buyers and can be differentiated by specific features that are valuable to customers who pay to buy them.

Services:

Services are other targeted objects of business transactions that can be differentiated in a variety of types such as information, brokerage, advertising, commerce support, utilities, networking, and personalized services.

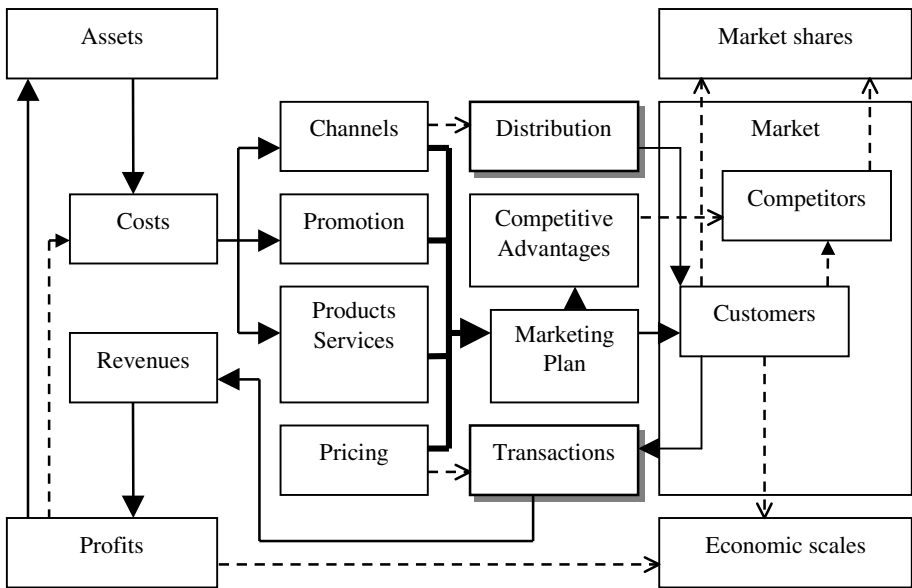


Fig. 1. A component structure of business models

Assets:

Assets, being one type of resources, are something with money values and are used as financial supports for companies to run businesses. Tangible assets consist of fixed and floating capitals such as equipment, cash reserves, stocks, and money raised through initial public offering (IPO). Other intangible assets include trademarks, brand awareness, technology infrastructure, patents, knowledge and expertise.

Costs:

Costs are necessary expenses for starting and continuing business operations including expenses and charges on products and services development, websites and information systems implementation, marketing, purchasing, inventory, distribution, transaction processing, human resources, Internet and other intermediary services, investment and acquisition, and goodwill amortization, etc.

Prices:

Prices are specified money values for customers to pay in exchange of products and services. Besides of the fixed pricing method, the frequently used dynamic pricing methods include negotiation, auction, and barter. Billing methods include charging the customers by volume or times, by month, by project, by transaction, or charging the advertisers.

Promotion:

Promotion is one of the marketing activities to capture customer attentions and to stimulate their buying desires. Possible types of promotion include advertisement, price discount, gift, trade show, and other related activities.

Distribution:

Distribution is an activity to deliver information, products and services through online and/or physical channels.

Revenues:

Revenues are incoming money received from prices paid by customers who buy products and services. Revenue sources include products, services, and advertising sales, as well as transaction fees, trading commissions, etc.

Profits:

Profits, reflecting company's business performance, are net earnings that equal to the difference between total revenues and total costs.

Market share:

Market share, representing portion of the market size owned by the company, is a percentage number obtained from dividing the company's volume of sales or size of customer body by that of the whole market.

Economic scale:

Economic scale is a target size of customer body for the company to break even and start gaining profits.

Marketing strategies and plans:

Marketing strategies and plans are strategic marketing decisions and action processes related to products, prices, promotions, and places factors, as well as their mix.

Competitive advantages:

Competitive advantages represent strength and capabilities of the company to outperform competitors by offering better values to customers, increasing market share, and sustaining profitability.

A company should start and run EC business with a soundly based assets to secure payments of costs incurred for developing, promoting, and delivering products and services. To assure profits, costs must be controlled carefully to keep under the level of revenues generated from selling products and services to the customers. By implementing a well-designed marketing plan, a company should be able to develop and offer high quality products and services to customers of the target markets with better values than that of its competitors. Properly planned promotion activities and pricing methods are implemented to increase the brand and product awareness and to gain competitive advantages. Customers can easily activate transactions to search and browse product-related information, place orders, pay the prices, and then receive the products and services through established sales and distribution channels. The company can then expect to attain high growth rates on revenues and market shares, and ultimately gain profits when the economies of scales are reached. Moreover, the company needs to properly plan the usage of assets and to raise enough capitals for covering all possible expenses and costs to ensure continuous business operations and to fulfill the goal of sustaining competitive advantages as well as profitability.

4 A Content Framework for EC Case Studies

Although there are lots of case studies related to EC applications, no common framework exists to standardize the content structure for comprehensive case studies as well as comparative business performance analyses. By investigating different views and structures of existing case studies for business companies, it can be found that the often presented content elements include background and history, target market and industry, business model and process, products and services, competition and key competitors, current situation and financial status, next steps and future directions, discussions and conclusions, etc. In most cases, the content described under the title of business model varied quite a lot due to the inconsistent meaning of business models conceived of by different case presenters. Business models are often treated as business strategies and processes, or sales and revenues. They are neither addressed by using an integrated view as stated in the previous section nor are arranged as the core of the case studies. As a result, it is not easy to realize the causal relations between business models and business performances, and it is even harder to perform comparative analyses based on these cases even if they actually belong to the same industry and adopt the similar models and processes. To improve the qualities and values of EC case studies, it is necessary to provide a common framework for organizing case contents in which the business model is included as a key part of the cases and can provide linkages to all other case-related content portions. Therefore, a feasible content structure of case studies should contain the following structural sections and content elements to provide a sound basis for supporting performance evaluation and comparison. A proposed content structure of case studies is shown in Figure 2.

1. **Business background and history:**
The very first part of the case study describes the background of the business, historical events and current status of the company including mission and goals, organizational structures, assets structure and capital conditions, etc.
2. **Products and services:**
This part describes the key characteristics and features of products and services the company offers to the markets, including customer value and competitor differentiation.
3. **Markets and Industry:**
In this portion, markets and associated segments, customers, competitors, supply chain participants are described, factors of competition and market shares are identified, and current status as well as problems and issues of the dynamic changing industries are addressed.
4. **System development process:**
This is to describe the costs and schedules for implementing EC information infrastructures including information systems and their functions, networks and communication facilities, business applications and customer supports, etc.

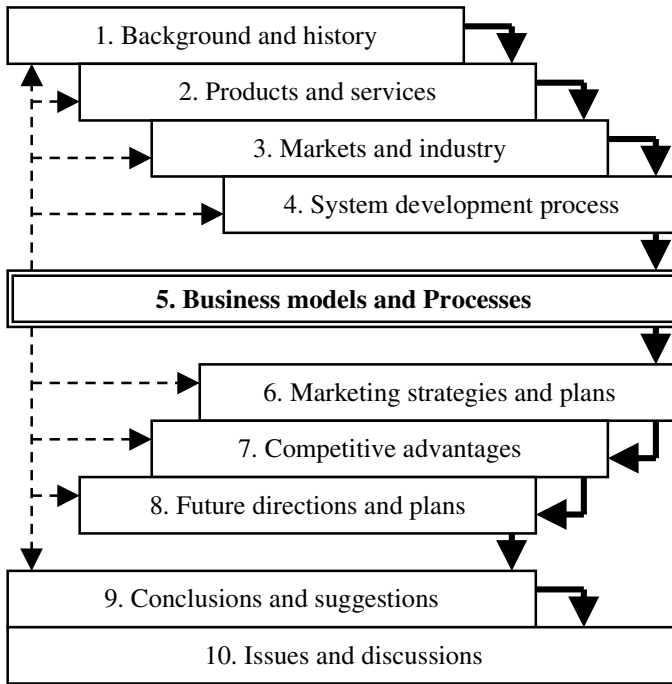


Fig. 2. A content structure of case studies

5. Business models and processes:

In this critical section of case studies, models of business operations and processes are presented and fully described. Major components include pricing and billing methods of products and services, cost structures and revenue sources, operating costs and revenues, transaction and payment processes, as well as profit conditions and economic scales.

6. Marketing strategies and plans:

Marketing strategies to be stated in this section contain strategic decisions and policies of brands and reputation, products and services, product positioning and market segmentation, prices, advertising and promotions, channels and distributions, as well as supply chain alliances, and affiliated programs. The Marketing plan, on the other hand, is an implementation plan of the marketing mix. Also stated are how can the marketing strategies and plans help building company reputation, increasing brand awareness, gaining larger market share, generating more revenues, maintaining competitive advantages, as well as ensuring and sustaining profitability.

7. Competitive advantages:

This part describes the company's distinctive capabilities and competencies that are resulted from implementing right business models and marketing plans, and thus can create competitor differentiation to outplay the competitors. SWOT (Strength, Weakness, Opportunity, Threat) analysis as well as Porter's

competitive forces model can also be included to help demonstrating how competitive edges can be increased.

8. Future directions and plans:

In this section, trends of the technology advancement and business evolution as well as the future developments of markets and industries are predicted and their impacts discussed. The short-term, mid-term, and long-term plans for continuing business operations, extending business scopes, sustaining profitability, achieving market leadership, etc are also described.

9. Conclusions and suggestions:

This part provides comprehensive conclusions on the company's models and performances accompanied by suggestions for making business improvement.

10. Issues and discussions:

This final portion of case studies lists problems and issues related to models, processes, applications, and management realized from the observed cases, and also offers some clear guidelines for further discussions.

5 Conclusions

Electronic commerce drives business enterprises to develop new ways of conducting business via the Internet and thus gives rise to new kinds of business models. Issues related to business models have quickly emerged as critical research topics in the EC domain. But in the literature, previous research works still lack of a consistent view and universal framework for defining and organizing business models. In this paper we present an integrated framework for illustrating key components and associated interactive relationships of business models. The proposed architectural framework aims at providing a common basis to unify diversified views on formulating, classifying and describing business models as well as to point out the critical factors for sustaining competitive advantages and profitability. Major components and elements mentioned include markets, customers, products, services, assets, costs, prices, promotion, distribution, revenues, profits, market shares, economic scales, marketing strategies and plans, and competitive advantages. Based on this model framework we also provide a content structure for case studies in which the business model is the core in order to provide a well-structured guideline for facilitating the conduction of sophisticate EC case studies and for supporting cross-business or cross-industry comparative analyses. This proposed content structure contains ten interrelated sections of content elements including business background and history, products and services, markets and industry, system development process, business models and processes, marketing strategies and plans, competitive advantages, future directions and plans, conclusions and suggestions, as well as issues and discussions. Future research works will focus on performing practical case studies based on the proposed framework and structures to validate the efficiency and effectiveness of our integrated approach as well as to demonstrate how this approach can support model classification, business process reengineering, strategic marketing decision making, and performance evaluation.

References

1. Afuah, A., Tucci, C. L.: Internet Business Models and Strategies: Text and Cases. McGraw-Hill (2001)
2. Boulton, R. E. S., Libert, B. D. A.: Business Model for the New Economy. *Journal of Business Strategy*. 21(4) (2000) 29-35
3. Clinton, W., Mateyaschuk, J.: What's the Model. *InformationWeek*. 745 (1999) 46-51
4. Enders, A., Jelassi, T.: The Converging Business Models of Internet and Bricks-and-Mortar Retailers. *European Management Journal*. 18(5) (2000) 542-550
5. Huff, S. L. et al.: Cases in Electronic Commerce. McGraw-Hill (2000)
6. Kleindl, B.: Competitive Dynamics and New Business Models for SMEs in the Virtual Marketplace. *Journal of Developmental Entrepreneurship*. 5(1) (2000) 73-85
7. Kogler, B., Lebowitz, J.: Integrating the e-Business Model. *Mortgage Banking*. March (2000) 66-74
8. Kraemer, K. L., Dedrick, J., Yamashiro, S.: Refining and Extending the Business Model with Information Technology: Dell Computer Corporation. *The Information Society*. 16(1) (2000) 5-21
9. Lee, J. K., Yang, C. Y.: Evolutionary Business Models of E-Cash with Smart Cards. *Proceedings of the International Conference on Electronic Commerce 2000*. Seoul, Korea (2000) 352-358
10. Mellahi, K., Johnson, M.: Does It Pay to be a First Mover in E.Commerce? The Case of Amazon.com. *Management Decision*. 38(7) (2000) 445-452
11. Muller, J. P., Pischel, M.: Doing Business in the Information Marketplace. *Proceedings of the Third Annual International Conference on Autonomous Agents*. Seattle, WA, USA (1999) 139-146
12. Rappa, M.: Business Model on the Web. <http://ecommerce.ncsu.edu/business.models.html>
13. Rayport, J. F.: The Truth About Internet Business Models. *Strategy+Business*. 16 (1999) 1-3
14. Schneider, G. P., Perry, J. T.: Electronic Commerce. Course Technology (2000)
15. Timmers, P.: Business Models for Electronic Markets. *Electronic Markets*. 8(2) (1998) 3-8
16. Turban, E., Lee, J., King, D., Chung, H. M.: Electronic Commerce: A Managerial Perspective. Prentice Hall (2000)
17. Westland, J. C., Clark, T. H. K.: Global Electronic Commerce: Theory and Case Studies. The MIT Press (2000)

Models and Protocol Structures for Software Agent Based Complex E-Commerce Transactions

Guandong Wang and Amitabha Das

School of Computer Engineering, Nanyang Technological University
Nanyang Avenue, Singapore, 639798
asadas@ntu.edu.sg

Abstract. The use of autonomous software agents enable new types of complex transactions for electronic commerce where multiple agents can exchange their values to complete a single transaction. However, most of the existing transaction protocols support only simple transactions and are not sufficient for more flexible and complex transaction scenarios. In this paper, we provide a complex transaction model which describes the complex transaction as a transaction tree. Based on this structure, a protocol structure is provided to ensure strong fairness, non-repudiability, timeliness and atomicity of the distributed complex transactions.

Key words: electronic commerce, transaction protocol, multi-agents

1 Introduction

Most of the widely used current e-commerce systems are based on electronic catalogs with a simple transaction model. This is quite a distance away from what an electronic marketplace is envisioned to be. Many essential functions from traditional commerce need to be effectively involved in e-commerce. Software agent technology enables new types of complex transactions in which multiple trading agents exchange their values in a distributed manner for a single transaction. Although a number of transaction protocols are existing [5, 13, 2, 7, 3] now, most of them are limited to simple exchange of funds and merchandise, and not sufficient for the complex transaction scenarios. Electronic commercial exchanges may be stymied because of a lack of a proper transaction model or protocol. The project of the Mixed Agent based E-Commerce System is started by our research group now with the objective to design and implement a software infrastructure for realizing a dynamic, decentralized and automated electronic market over the Internet. In this paper, we aim to propose a complex transaction model for electronic commerce that will facilitate the growth of successful transactions on the Internet and provide a complex transaction protocol structure according to this model.

2 Simple Transactions

In the e-commerce world, the simplest way for the customers is to buy the products from the manufacturers' web site, such as "Dell online store"[4]. We

call it the “*Direct Transaction Model*” (*DTM*). However, not all manufacturers have the abilities to effectively push out their products through their own web sites. So some electronic shops (e-shop) come into being, such as Amazon.com[9]. These e-shops pre-purchase products from the manufacturers with their own funds and sell those products through their web sites. We call this model the “*Retailer Transaction Model*” (*RTM*). However, the RTM may cause the e-shop companies run the risk of being stranded with unsold stock if they can not sell the products purchased from manufacturers. A directory oriented model resolves this problem. In this, the e-shop only provides directory services which help the customer find the desired products and the corresponding manufacturers or dealers. Once the customer finds the favorable products, he just buys directly from the manufacturer and the e-shop is not involved in the transaction process. We call this model the “*Directory Transaction Model*” (*DITM*). One example of this model is eBay[6].

All the above three transaction models represent one-to-one transactions, i.e., transactions are made directly between two parties: one is the customer who needs the products and the other is the supplier who owns the products. We define this kind of transactions as *simple transactions*.

3 Complex Transactions: Definitions and Models

The DITM removes the pre-purchase process in the RTM and minimizes the risks of losing money for the e-shops. There is still another solution. We can change the pre-purchase process to be part of the electronic transactions. After the e-shop receives the demands from the customers, it passes the requirements to the corresponding sellers. If both of the customer and the seller are willing to make a deal, the e-shop first buys the products from the seller and then resells them to the customer. We call the e-shop in this scenario a broker.

Actually the seller here does not need to be a supplier. Another broker can also advertise to the broker. This results in an end to end transaction Chain. We call this model the “*Chained Transaction Model*” (*CTM*). In this model, theoretically unlimited parties can participate in the whole transaction. Among these parties, there should be a customer who is an end buyer and a supplier who is an end seller. The other parties are brokers who buy and resell the products.

Sometimes, the customer may want to purchase a number of products together from different sellers, and an incomplete combination of the part products have no value to the customer. As a result, the customer is only willing to purchase all the products or none. We call this model the “*Aggregate Transaction Model*” (*ATM*). This kind of transaction has been mentioned in [11] as an infeasible transaction scenario.

Sometimes, the customer may want to purchase exactly one product from one of several possible sellers. For expediency, the customer may proceed with multiple transaction options simultaneously, but in the end exactly one of them should be committed. We call this model the “*Optional Transaction Model*” (*OTM*)

In the physical commerce world, it is very common for a manufacturer to bundle some other companies' products or services such as advertising, marketing, packing or delivering to his own products. In the e-commerce world, this kind of bundling is always achieved through B2B e-commerce with a relatively static contract. A more flexible approach could be dynamically to search for a company and purchase its products or services in a transaction process. We call this model the “*Bundled Transaction Model*” (BTM).

Unlike the simple transactions, these four kinds of transactions are not just between two parties. The whole transaction consists of several sub-transactions. This brings more complexities and more transaction issues.

3.1 A Formal Definition of Complex Transaction

A complex electronic commerce transaction is a collection of mutually related operations executed among several parties through networks, for exchanging goods or services for money between one customer and some suppliers. There can be many intermediaries involved in a transaction, but we use the term *customer* to identify the party who is the final recipient of the goods. Similarly, we use the term *supplier* to indicate the original source of the goods in a given transaction. One transaction process must have one customer with some requirements for goods or services, and at least one supplier with some physical or abstract products. The transaction is considered to be completely successful, when the customer pays for and gets the products which satisfy his or her expectations from the transaction.

Definition 1 (Complex Electronic Commerce Transaction). *Given a set of transaction parties $P = \{p_1, p_2, \dots, p_n\}$, and a set of products $V = \{v_1, v_2, \dots, v_l\}$, a complex e-commerce transaction can be defined as a Directed Tree (N, E) where:*

- (1) $N = M \cup L$ is a set of nodes. The elements of $M = \{m_i | i = \{1, 2, \dots\}, m_i \in P\}$ are called **member-nodes**. Each member-node represents a transaction party involved in the transaction. The elements of $L = \{l_i | i = \{1, 2, \dots\}, l_i \in \{\vee, \wedge\}\}$ are called **logic-nodes**. There are two types of logic-nodes. We call the nodes of type ‘ \vee ’ **exclusive-nodes** and the nodes of type ‘ \wedge ’ **inclusive-nodes**.
- (2) $E = E_1 \cup E_2$ is a set of edges. The elements of $E_1 = \{e_{a_i \rightarrow b_i : v_i} | i = \{1, 2, \dots\}, a_i, b_i \in M \cup L, a_i \neq b_i, v_i \subset V, v_i \neq \phi\}$ are called **value-transfer-edges**. Each of these edges means that some values represented by v_i are transferred from a_i to b_i . $e_{a_i \rightarrow b_i : v_i}$ is called **in edge** for b_i and **out edge** for a_i . v_i is called **in value** for b_i and **out value** for a_i . If $B = \{b_i | i = \{1, 2, \dots\}\}$ and $A = \{a_i | i = \{1, 2, \dots\}\}$, the set $B \setminus A$ contains only one node. This node is the root of the tree and represents the **customer** for the transaction. The nodes in $B \cap A$ are defined as **brokers** for the transaction. The elements of $E_2 = \{e_{\phi \rightarrow s_i : v_i} | i = \{1, 2, \dots\}, s_i \in M, v_i \subset V, v_i \neq \phi\}$ are called **value-input-edges**. $e_{\phi \rightarrow s_i : v_i}$ is called **in edge** for s_i . v_i is called **in value** for

s_i . The nodes in $S = \{s_i | i = \{1, 2, \dots\}\}$ are called the **suppliers** for the transaction. E_2 is a nonempty set.

- (3) All the member-nodes of the transaction except the customer have exactly one in value and one out value. And the in value is equal to the out value.
- (4) For the logic nodes, there are multiple in-edges and one out edge. For the inclusive-node, the out value is equal to the conjunction of the in values, represented by (\cdot, \cdot, \cdot) . For the exclusive-nodes, the out value is equal to the disjunction of the in values, represented by $< \cdot, \cdot, \cdot >$.

The definition describes a generic complex electronic commerce transaction process. According to this definition, Figure 1(a) is a complex transaction. From this generic definition we can propose more specific definitions for the representative transaction models(see Fig. 1) as follows:

Definition 2 (Simple Transaction). A simple transaction can be defined as $(\{a, b\}, \{e_{a \rightarrow b:v}, e_{\phi \rightarrow a:v}\})$ where a is a supplier and b is a customer.

The simple transaction has just one operation in which product v is transferred from the only supplier a to the only customer b .

Definition 3 (Chained Transaction). A chained transaction can be defined as $(\{b_1, b_2, \dots, b_j, c\}, \{e_{\phi \rightarrow b_1:v}, e_{b_1 \rightarrow b_2:v}, \dots, e_{b_{j-1} \rightarrow b_j:v}, e_{b_j \rightarrow c:v}\})$ where c is a customer, b_1 is a supplier and $b_i (i = 2, \dots, j)$ are brokers. (See Fig. 1(b))

The chained transaction consists of two or more operations. The first one transfers the product v from the supplier b_1 to the broker b_2 , and the last one transfers the product v from the broker b_j to the customer c . The whole transaction operates on one product. The total effect is transferring the product v from b_1 to c through a set of brokers (b_2, \dots, b_j) one by one.

Definition 4 (Aggregate Transaction). An aggregate transaction can be defined as $(\{c, a_1, a_2, \dots, a_j, l\}, \{e_{\phi \rightarrow a_1:v_1}, e_{\phi \rightarrow a_2:v_2}, \dots, e_{\phi \rightarrow a_j:v_j}, e_{a_1 \rightarrow l:v_1}, e_{a_2 \rightarrow l:v_2}, \dots, e_{a_j \rightarrow l:v_j}, e_{l \rightarrow c:(v_1, v_2, \dots, v_j)}\})$ where $a_i (i = 1, 2, \dots, j)$ are suppliers, c is a customer, l is an inclusive node, and (v_1, v_2, \dots, v_j) represents the conjunction of the products $v_i (i = 1, 2, \dots, j)$. (See Fig. 1(c))

The aggregate transaction involves multiple products. All the suppliers transfer their products to the customer simultaneously. The inclusive node in the definition signifies that the customer will consider the transaction to be successful if and only if all the values have been successfully transferred from the suppliers to the customer.

Definition 5 (Bundled Transaction). A bundled transaction can be defined as $(\{a, b, c\}, \{e_{\phi \rightarrow a:v_1}, e_{a \rightarrow b:v_1}, e_{\phi \rightarrow b:v_2}, e_{b \rightarrow c:(v_1, v_2)}\})$ where a and b are suppliers, and c is a customer. (See Fig. 1(d))

In the bundled transaction, the supplier transfers his product as well as a product purchased from another supplier to the customer.

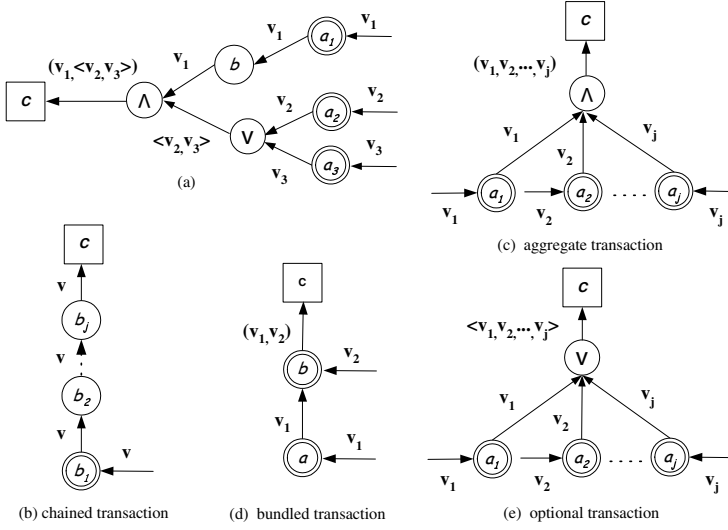


Fig. 1. Example of Some Specific Transactions

Definition 6 (Optional Transaction). *An optional transaction is defined as $(\{c, a_1, a_2, \dots, a_j, l\}, \{e_{\phi \rightarrow a_1:v_1}, e_{\phi \rightarrow a_2:v_2}, \dots, e_{\phi \rightarrow a_j:v_j}, e_{a_1 \rightarrow l:v_1}, e_{a_2 \rightarrow l:v_2}, \dots, e_{a_j \rightarrow l:v_j}, e_{l \rightarrow c:\langle v_1, v_2, \dots, v_j \rangle}\})$ where $a_i (i = 1, 2, \dots, j)$ are suppliers, c is a customer, l is an exclusive node, and $\langle v_1, v_2, \dots, v_j \rangle$ represents one of the products $v_i (i = 1, 2, \dots, j)$. (See Fig. 1(e))*

In the optional transaction, the customer c tries to purchase a product from one of multiple suppliers. The exclusive node in the definition means that the customer will consider the transaction to be successful if and only if one of the products has been successfully transferred from one supplier to the customer and all the other operations have been safely aborted.

The above four kinds of transactions are some special cases of complex e-commerce transactions. Each of them has a directed tree structure. From this tree structure, we can derive another tree structure representing the hierarchy among the sub-transactions constituting a single complex transaction. We define the whole transaction tree as the *top-level transaction*. Then any sub-tree, whose leaf nodes and root are all member nodes, is defined as a *sub-transaction* of the top-level transaction. If the buyer of a sub-transaction A is the seller of another sub-transaction B , then the sub-transaction A is B 's *parent* and the sub-transaction B is A 's *child*. We define an *operation* as a sub-transaction with only two member nodes. In other words, an operation corresponds to a component simple transaction, in which the seller sells the products to the buyer.

4 Requirements of E-Commerce Transactions

Researchers have analyzed the requirements for e-commerce transactions from different aspects. Asokan specified the fairness to be the basic requirement for e-commerce transactions[1]. Tygar applied the atomicity concept into each crucial aspect of e-commerce transactions[12]. However, all the above works focused on what we have defined as simple transactions. We use and extend the above ideas to encompass complex transactions and analyze the requirements of complex transactions that involve more than two parties and multiple sub-transactions defined earlier. We identify four requirements for a successful e-commerce transaction, that are mutually orthogonal and must be addressed explicitly in the design of the protocol. These are, atomicity, Fairness, Non-repudiability and Timeliness. We discuss each of these requirements below.

4.1 Atomicity

As is well-understood, atomicity refers to an all-or-none semantics, that is, a transaction should be completed either in its entirety or not at all. In the complex transactions, there are some logic relations among the operations or the sub-transactions. The atomicity in a complex transaction means that either all the operations in the transaction are successfully executed according to the internal logic, or none of them is executed. This implies, for example, that two exclusive operations can not both be committed if the transaction is atomic. The atomicity of a complex transaction mainly aims to avoid the following situations at the end of the transaction:

- **Risk One:** The broker is stranded with useless goods which he has purchased from the sellers but can not resell to the buyer. This case may arise in chained or bundled transactions.
- **Risk Two:** The buyer who required a combination of products, gets and pays for only a part of them which has no value to him. This case may arise in aggregate transactions.
- **Risk Three:** The buyer who had intended to purchase one product from one of several possible sellers, ends up paying for more than one products which satisfy the same requirement of him. This case may arise in optional transactions.

4.2 Fairness

Fairness is the basic requirement for e-commerce transactions. In our approach, a *fair exchange protocol* requires that during the exchange process, one player will not gain any advantage over the other player at any step of the exchange protocol. This implies that even if an unexpected failure happens while the exchange protocol is in progress, neither of two players gains more value over the other. The fairness of a complex transaction mainly aims to avoid the following situations:

- The buyer has paid but has not got the products for a long time.
- The seller has sent the products but has not got the payment for a long time.
- The exchange has been canceled, but the buyer was charged any way.
- The exchange has been canceled, but the seller's products have been used.
- The exchange has been canceled, but the seller can not get back his products.

4.3 Non-repudiation

Non-repudiation is another essential property required for e-commerce transactions. The main purpose of a non-repudiation service is to protect the parties involved in a transaction against the other party denying a particular event or action after the transaction process.

The International Standardization Organization (ISO) has standardized techniques to provide non-repudiation services in open networks. One version of the draft ISO standards [10] identifies various classes of non-repudiation services. Two of these are of particular interest. *Non-repudiation of Receipt (NRR)* guarantees that the recipient of a message cannot deny having received that message. *Non-repudiation of Origin (NRO)* guarantees that the originator of a message cannot later deny having originated that message. In the case of complex transactions, non-repudiation should be able to resolve the following problems.

Non-repudiation of Origin:

- The buyer says, “This isn’t the product I specified.”
- The seller says, “I’ve received payment, but it is faked money.”
- The seller says, “I should be paid \$X, but only received \$Y.”

Non-repudiation of Receipt:

- The buyer says, “I’ve paid, but he says I have not.”
- The seller says, “I’ve sent the product, but he says he has not received it.”

4.4 Timeliness

It is a reasonable requirement for any e-commerce transaction that the transaction should be completed within a certain time which may be specified by the buyer. In some events, time is a crucial factor for the customer. Before accepting the payment, the seller must have the confidence of being able to deliver the product to the buyer on time. If the buyer can not receive the product on time, he should be able to get back his money.

5 The Complex Transaction Protocol

The requirements described in the previous section have been addressed in the past in the protocols designed for what we termed as ‘simple transactions’. Though the requirements of complex transactions are similar to those of simple transactions, the mechanisms for ensuring them are expected to be different and more complex. In what follows, we lay out the basic properties of these mechanisms that essentially determine the structure of the protocols to be used for complex transactions.

5.1 Mechanism to ensure atomicity

The three risk categories described earlier under the section on atomicity result from the non-atomic execution of the transaction where only some of the sub-transactions are committed according to the logic while other sub-transactions are unexpectedly aborted or unexpectedly committed. In order to avoid these risks, we set down the rules of commit and abort dependencies for complex e-commerce transactions as follows:

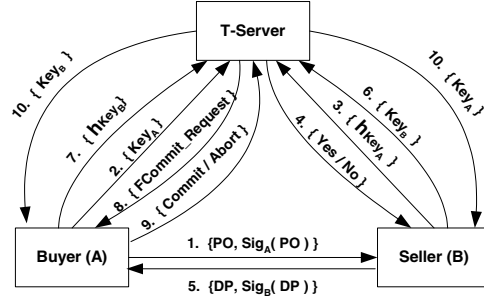
1. Commit Dependencies for Complex Transactions: the parent transaction cannot commit until all its children commit ¹. This rule aims to protect the buyers in aggregate transactions against purchasing ‘half’ products.
2. Abort Dependencies for Complex Transactions: all the children should be aborted if the parent transaction aborts. This rule removes the first risk that is applicable to the chained and bundled transactions. It protects the brokers from losing money on not being able to resell the products they’ve purchased.
3. Commit Dependencies for Optional Transactions: an optional transaction can not commit until one of its children commits. This implies that if all of its sub-transactions are aborted, the optional transaction should be aborted.
4. Abort Dependencies for Optional Transactions: if any one of the children of an optional transaction commits, all the other children should be aborted. This rule eliminates the third risk of complex transactions and protects buyers from wasting money on duplicate products.

For ensuring the atomicity of the complex transactions, we define two commit phases for each operation. We define the *locally committed* status as the state when all the items required for completing the transaction have been stored to a reliable hardware and operation is able to make the transaction finally committed or safely aborted at any time. We further define the *finally committed* status as the state when both the buyer and the seller have received the mutually agreed upon goods and money respectively. A transaction is ready for final commit only when all the required sub-transactions have been locally committed. The mechanism to ensure atomicity will be presented in somewhat more detail later in the protocol structure section.

5.2 Mechanism to ensure fairness and non-repudiation

The requirements of fairness and non-repudiation must be satisfied for each of the component sub-transactions of a complex transaction. On the other hand, any solution that guarantees strong fairness must use a trusted third party (TTP) [8]. As a result, a fair complex transaction protocol must involve three parties in each simple sub-transaction: a buyer, a seller and a trusted third party represented by a transaction server (T-Server). Before each simple transaction, the buyer and the seller must first agree on a transaction server.

¹ Here, if the parent transaction is an optional transaction, then the transaction should follow the third rule.



$$PO = \{E_{Key_A}(EP), hKey_A, T, a, desc, t, V_A\}$$

$$DP = \{E_{Key_B}(G), hKey_B, T, a, desc, V_B\}$$

Fig. 2. The Exchange Protocol

The mechanism to ensure fairness and non-repudiation is mostly the same as other fair exchange protocols such as [13]. But, the protocol for the exchange operations in the complex transactions needs a locally committed stage. Here, the T-Server is used to control the commitment status in each operation of the complex transaction. For each exchange operation, the buyer and the seller first send their items to the agreed T-Server. When the T-Server has stored all the items required for completing the exchange to its reliable local hardware, the operation is locally committed. From this time, neither the buyer nor the seller can abort or finally commit the exchange without the help of the T-Server.

Figure 2 shows the exchange protocol for the operations in the complex transaction. PO is a payment order. DP is a delivery package. EP is the electronic payment. G is the digital goods or a delivery order. T represents the agreed T-Server. ' a ' equals to the amount of the payment. $desc$ is the description of the product. ' t ' is the time limitation for the transaction. The operation is considered to be locally committed after step 7. For the sake of space limitation, we can not describe the protocol in detail in this paper.

5.3 Mechanism for ensuring timeliness

Timeliness of an e-commerce transaction is essentially defined by the customer. Therefore, as long as the customer is able to decide just before finally committing whether the transaction satisfies the requirement of timeliness, and he is free to reject or abort the transaction if unsatisfied, timeliness is ensured automatically without imposing any overhead on the protocol.

5.4 The Protocol Structure

Any fair exchange protocol involving two parties (and the TTP) has two phases² with the following semantics:

Payment Phase: In this phase, the encrypted payment or a non-retractable guarantee of payment flows from the buyer to the seller, with the TTP acting as the guarantor. The payment guarantee, is, however, conditioned on the fact that the buyer must receive the goods according to the specifications agreed upon.

Delivery Phase: In this phase, the flow of goods takes place from the seller to the buyer. Once the TTP is satisfied that the customer has received the specified product, it transfers the payment to the seller. In this case, the transaction can be considered to be finally committed as soon as the seller delivers appropriate goods to the buyer.

In the case of complex transactions, any protocol for simple transactions that ensures fairness and non-repudiation can be adopted as the core protocol for the component sub-transactions. The above protocol structure is essentially maintained for each of the sub-transactions, with some modification in the semantics. In addition, a third phase needs to be added in order to ensure the atomic property of the complex transaction. The transaction starts from the top-level sub-transaction (root). The whole transaction completes in three phases which are outlined below:

1. **Payment Phase.** In this phase, payment guarantee flows from the buyers to the sellers (or from the root to the leaves). This phase has similar structure as the corresponding phase in a simple transaction. The only difference lies in the condition associated with the payment guarantee. The modified condition for the payment guarantee to materialize is that the seller must provide delivery guarantee, and that the buyer must endorse the transaction at the final phase.
2. **Delivery Phase.** In this phase, a conditional delivery guarantee flows from the sellers to the buyers up the transaction tree, with the corresponding TTP as the guarantor at each level. At the end of the delivery phase, a sub-transaction is locally committed, because, after this point onwards, the seller has no way to reject a transaction if it is subsequently finally committed by its superiors.
After all the children of the top-level transaction have locally committed, the customer can decide to finally commit the transaction, and then comes the third phase.
3. **Final Commit Phase.** In this phase, the (sub-)transactions are finally committed from the root to the leaves. For each operation, the relevant TTP converts the guarantee into actual payment or delivery, as the case may be.

6 Conclusion and Future Work

Current status of e-commerce is still unsatisfactory. The issues mostly concern information filtering, real time negotiating and secure transacting. Among these

² These two phases are not necessary ordered as they are presented below.

three aspects, the transaction issue is the most critical since the transaction process represents the real value of e-commerce. However, most of the existing e-commerce transaction systems are based on web catalogues strategy. Though a few agent-based e-commerce systems provide some improvements, they provide no practical transaction schemes.

The complex transaction model developed in this paper is a significant improvement over the previous works as it more effectively captures some of the complexities of real-world transactions. This model and the protocol structure in turn, will pave the way for sophisticated e-commerce activities based on autonomous agents. We are interested in the agent based transaction system since we believe that it is the only way to bring enough flexibility and liveness into the e-commerce world.

References

1. N. ASOKAN. *Fairness in electronic commerce*. PhD thesis, University of Waterloo, May 1998.
2. N. ASOKAN AND M. SCHUNTER AND M. WAIDNER. Optimistic protocols for fair exchange. In *Proceedings of 4th ACM Conference on Computer and Communications Security*, Zurich, April 1997.
3. BAO, F. AND R. DENG AND W. MAO. Efficient and practical fair exchange protocols with off-line ttp. In *1998 IEEE Symposium on Security and Privacy, IEEE Compute Society*, pages 77–85, Oakland, May 1998.
4. DELL COMPUTER CORPORATION. <http://www.dell.com/>.
5. BENJAMIN COX AND J.D. TYGAR AND MARVIN SIRBU. Netbill security and transaction protocol. In *The First USENIX Workshop on Electronic Commerce*, pages 77–88, July 1995.
6. EBAY INCORPORATION. <http://www.ebay.com/>.
7. M. K. FRANKLIN AND M. K. REITER. Fair exchange with a semi-trusted third party. In *Proceedings of The Fourth ACM Conference on Computer and Communications Security*, pages 1–6, Zurich, April 1997.
8. HENNING PAGNIA, FELIX C. GARTNER. On the impossibility of fair exchange without a trusted third party. Technical report, Department of Computer Science, Darmstadt University of Technology, March 1999.
9. AMAZON.COM INCORPORATION. <http://www.amazon.com/>.
10. ISO/IEC JTC1, INFORMATION TECHNOLOGY SC 27. Information technology - security techniques - non-repudiation. Technical report, ISO/IEC JTC 1/SC 27, 1997. Contains 3 parts; Current version dated, March 1997.
11. STEVEN PAUL KETCHPEL. *The Networked Information Economy: Applied and Theoretical Frameworks for Electronic Commerce*. PhD thesis, Stanford University, August 1998.
12. J.D. TYGAR. Atomicity in electronic commerce. In *Proceedings of the the Fifteenth Annual ACM Symposium on Principles of Distributed Computing*, pages 8–26, May 1996.
13. JIANYING ZHOU AND DIETER GOLLMANN. A fair non-repudiation protocol. In *Proceedings of the IEEE Symposium on Research in Security and Privacy [IEEE96]*, pages 55–61, 1996.

A Multidimensional Approach for Modelling and Supporting Adaptive Hypermedia Systems

Mario Cannataro, Alfredo Cuzzocrea, and Andrea Pugliese

ISI-CNR, Via P. Bucci, 41/c
87036 Rende, Italy
{cannataro, apugliese}@si.deis.unical.it,
cuzzocrea@isi.cs.cnr.it

Abstract. In this paper we present an XML-based modular architecture for the modelling and the run-time support of web-based Adaptive Hypermedia Systems. The proposed model and the supporting architecture allow the hypermedia adaptation along three different “adaptivity dimensions”: *user’s behaviour* (preferences and browsing activity); *technology* (network, user’s terminal, etc.); *external environment* (time, location, language, socio-political issues, etc.). A view over the Application Domain corresponds to each possible position of the user in the “adaptation space”. For the description of the logical structure of the hypermedia we propose a graph-based layered model: XML-based models are used to describe metadata about basic data fragments and “neutral” pages to be adapted (i.e. presented) with respect to the user position. An authoring and simulation tool which allows the design and simulation of the Application Domain is also presented.

1 Introduction

The linking mechanism of hypermedia offers users a large amount of navigational freedom so that it becomes necessary to offer support during navigation. The personalisation of web presentations and contents, i.e. their adaptation to user requirements and goals, is becoming a major requirement of modern web-based systems. Different application fields where contents personalization can be useful are on-line advertising, direct web-marketing, electronic commerce, on-line learning and teaching, etc.

Main aspects that should be taken into account are:

- the different classes of users that will use the system; they are becoming more and more heterogeneous due to different interests and goals, world-wide deployment of services, social conditions, etc.;
- the kind of user terminals and network. User terminals can differ not only at the software level (browsing and elaboration capabilities), but also in terms of ergonomic interfaces (scroll buttons, voice commands, etc.); networks can differ e.g. regarding communication bandwidth (narrow, medium, broad) and dynamic properties (per-user bandwidth, latency, error rate, etc);
- the spatial and temporal conditions of the particular user, geo-political issues, etc.

To face some of these problems, in the last years the concepts of user modelling and adaptive graphical user interface have come together in the Adaptive Hypermedia (AH) research theme [2, 5]; some recent Adaptive Hypermedia Systems (AHS) are outlined in [5, 7].

Basic components of AHS are:

- the *Application Domain Model*, used to describe the hypermedia basic contents and their organisation to depict more abstract concepts. The most promising approach in modelling the Application domain is data-centric, and many researches employ well known database modelling techniques [1];
- the *User Model (profile)*, which attempts to describe the user's characteristics and preferences and his/her expectations in the browsing of hypermedia. In the following, we will use the term User Model to indicate the behavioural model only; the overall model, comprising non-behavioural characteristics, will be captured by means of the presented multidimensional approach;
- the *techniques to adapt presentations* with respect to the user's behaviour and to the content provider's goals. Such techniques can be generally distinguished into *adaptive presentation*, i.e. a manipulation of information fragments, and *adaptive navigation support*, i.e. a manipulation of the links presented to the user. More recently, the capability to deliver a given content to different kind of terminals, using e.g. wired or wireless networks, (i.e. the support of multi-channel accessible web systems), is becoming an important requirement of AHS.

In this paper we present an XML-based modular architecture for the modelling and the run-time support of web-based Adaptive Hypermedia Systems, supporting in particular multi-channel access.

The proposed model, named *XML Adaptive Hypermedia Model* (XAHM), allows the hypermedia adaptation along three different "adaptivity dimensions": *user's behaviour*, *technology* and *external environment*. The construction of a User Model, i.e. the assignment of the user's position over the user's behaviour dimension, is carried out using a probabilistic interpretation of hypermedia structure. Furthermore, XAHM makes use of XML technologies to (i) describe basic multimedia data, (ii) describe "neutral" pages to be adapted, (iii) offer an advanced multi-channel access support at run-time.

The rest of the paper is organised as follows. Section 2 presents XAHM. Section 3 describes the system architecture. Section 4 discusses related works. Finally, Section 5 contains conclusions and outlines future work.

2 Application Domain Modelling

This Section presents our approach to the modelling of Adaptive Hypermedia. After a description of the proposed adaptation scheme, we show a graph-based layered model for the Application Domain and the XML-based model used to describe metadata about basic information fragments.

2.1 Adaptation Space

The goal of AHSs is to adapt contents and presentations to satisfy the user's goals and/or requirements. In XAHM, the Application Domain is modelled along three different orthogonal adaptivity dimensions (Fig. 1):

- *User's behaviour* (preferences and browsing activity);
- *External environment* (location, language, socio-political issues, external web sites content, etc.);
- *Technology* (network bandwidth, Quality of Service, user's terminal, etc).

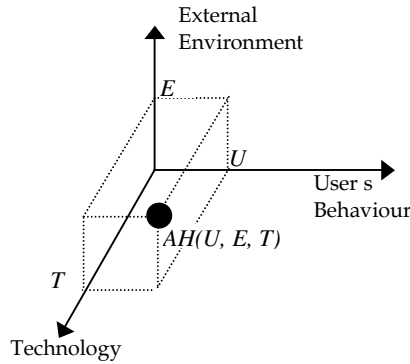


Fig. 1. Adaptivity dimensions and adaptation space

Each position of the user in the adaptation space is a tuple $[U, E, T]$ indicating that the user belongs to the group identified by the U value, is located into the environment identified by the E value, and is using the technology identified by the T value. The Adaptive Hypermedia is a sub-space $AH(U, E, T)$ of the overall adaptation space, specified by the author by specialising the description of the hypermedia for some particular positions of the user. A personalised view over the Application Domain corresponds to each of these possible positions, and is obtained instantiating the neutral XML descriptions of pages with respect to the actual position of the user.

The AHS monitors the different possible sources that can affect the position of the user in the adaptation space, collecting a set of values, called *User*, *Technological* and *Environment Variables*. The User and Environment variable mainly drive the generation of page content, while the technology one mainly drives the adaptation of page layout. The current position of the user along a generic dimension is obtained using a mapping function from the corresponding variables to a mono-dimensional value. For example, if we consider n Technological Variables, each of which has an associated domain V_i , $i=1, \dots, n$, then there will exist a simple mapping function $f: V_1 \times V_2 \times \dots \times V_n \rightarrow T$, and T will have $|V_1| \times |V_2| \times \dots \times |V_n|$ possible values. Without loss of generality, the domain T can be N .

A classification algorithm, whose description can be found in [7], carries out the mapping from the User Variables to the User Dimension (i.e. user profile).

2.2 A Layered Model for the Logical Structure of the Hypermedia

The proposed Application Domain Model uses a layered data model; it extends the Adaptive Data Model described in [9] and comprises the following levels:

0. *Information Fragments (IF)* or *atomic concepts*, like texts, sounds, images, videos, etc. at the lowest level. Data can be structured, semi-structured or unstructured and can be provided by different sources (e.g. external or local databases, XML and HTML documents, texts, files and so on). They are described by metadata represented by XML documents; the structure of such documents will be shown in Section 2.3.
1. *Presentation Descriptions (PD)*, which describe *Page Concepts* constituted of XML documents. A Presentation Description is a “neutral” page whose elements are be associated to a portion of the adaptation space so it is transformed and delivered to the user considering his/her position in that space. Each link in a page is annotated by weight, that represents its importance with respect to each other links, and that in our model is a probability (see Section 2.2.1). Final pages composed of actual fragments, called *Presentation Units (PU)* or *pages*, are dynamically generated at run time in a target language (XML, HTML, WML, etc.) and delivered.
2. *Elementary Abstract Concepts (EAC)* representing larger units of information. One or more Presentation Descriptions organised in a weighted digraph [11], compose Elementary Abstract Concepts. Arcs represent relationships between elementary concepts or navigation requirements.
3. *Application Domain*. Finally, an Application Domain (i.e. AH) is composed by a set of Elementary Abstract Concepts organised in a digraph. Arcs represent relationships between EACs. It should be noted that although the structures of the hypermedia data at the levels 2 and 3 of our data model are essentially the same, the semantic interpretation of the inter-EAC browsing is different with respect to the intra-EAC browsing.

The modelling of an adaptive hypermedia comprises the following phases:

1. definition of *M stereotype user profiles*, representing users’ groups;
2. definition of the overall Application Domain, as a directed graph of EACs, differentiating its link structure with respect to user profiles;
3. definition of the structure of each EAC, differentiating its link structure with respect to user profiles;
4. construction of the Presentation Descriptions, differentiated with respect to all the adaptivity dimensions.

The Application Domain and the EACs are differentiated with respect to the user’s profile only (User’s Behaviour dimension) because they need to be directly described by the author and it could not be suitable to build and maintain a large variety of weighted digraphs. However, future extensions of the system could support the personalization of the link structure of the hypermedia with respect to the Technology dimension, so allowing “lighter” (i.e. with shorter paths) versions of the hypermedia, to be browsed in a more agile way.

2.2.1 Probabilistic Graph-Based Scheme and User Classification

The proposed graph-based model supports a probabilistic interpretation of the arcs' weight – the mapping from the collected variables about user's behaviour (User Variables) to the corresponding dimension of the adaptation space (which, as said before, is organised identifying groups of users) is based on the evaluation of values related to such probabilistic interpretation. The mapping is carried out also considering some other values, related to *intrinsic properties* of the hypermedia structure.

An AD with M different user profiles can be seen as a “flat” set of XML documents where the generic document contains, for each profile, a set of annotated outgoing links. The AD can be mapped in a weighted digraph G where each node corresponds to a XML document and each directed arc to an outgoing link; in turn, the weighted digraph G can be referred to as the set of the weighted digraphs G_k , $k=1, \dots, M$, obtained extracting from G the nodes and arcs corresponding to each profile. Each G_k is named *Logical Navigation Graph* [6].

The proposed probabilistic approach assumes that the weight of the arc (i, j) in G_k is the conditional probability $P(j|k, i)$, namely the probability that a user belonging to the group k follows the link to the j node having already reached the i node.

The User Variables contain the recently followed path $R = \{R_l, \dots, R_{r-1}, R_r\}$ composed by the last visited nodes (where R_{r-1} is the current node and R_r is the next node) and the time spent on recent nodes, $t(R_l), \dots, t(R_{r-1})$, so identifying a sort of sliding temporal window of length r .

On this basis, the system evaluates, for each profile k , the following values:

- the probability of having followed the R path through arcs belonging to the profile k ;
- the “reachability” of the next node R_r starting from the first node R_l , through arcs belonging to the profile k (i.e. the maximum probability path in G_k);
- the distribution with respect to the profile k of the visited nodes from R_l to R_{r-1} , weighted with the time spent on each of them. For example, let $\{n_1, n_2, n_3\}$ be the recently visited nodes and $\{t_1, t_2, t_3\}$ the time units spent on each of them: if node n_1 belongs to profiles k_1 and k_2 , node n_2 belongs to k_2 and k_3 and node n_3 belongs to k_1 and k_4 , the distribution is evaluated as $[(k_1, t_1+t_3), (k_2, t_1+t_2), (k_3, t_2), (k_4, t_3)]$. The visiting times can be measured using a software running on the user terminal.

The intrinsic properties of the hypermedia are expressed, for each profile k , by:

- the average value of the probability of the minimum paths in G_k ; high values of this term indicate the existence of highly natural paths in the hypermedia;
- the average value of the length of the minimum paths in G_k ; high values of this term mean longer natural paths in the hypermedia, which could be an advantage in the overall personalization process;
- the number of nodes belonging to profile k .

The algorithm for the evaluation of the user's profile computes a “belonging probability” density function considering initial user choices (e.g. expressed in a questionnaire) and the aforementioned values; a detailed description of the algorithm is beyond the scope of this paper and can be found in [7].

2.3 XML Metadata about Basic Information Fragments

In XAHM, each data source is “wrapped” by an XML meta-description. The use of metadata is a key aspect for the support of multidimensional adaptation; for example, an image could be represented using different levels of detail, formats or points of view (shots), whereas a text could be organised as a hierarchy of fragments, represented using different languages. In the construction of pages, the author refers only to such metadata, avoiding too low-level access functions to real data fragments.

A number of *Document Type Definitions* [15] for the XML meta-descriptions have been designed; as an example, in Figure 2a is shown the meta-description of a relational table (or view) stored in a relational database.

<pre> <!ELEMENT table (column+)> <!-- ATTLIST table alias CDATA #REQUIRED IP-address CDATA #REQUIRED name CDATA #REQUIRED database-name CDATA #REQUIRED --> <!ELEMENT column EMPTY> <!-- ATTLIST column name CDATA #REQUIRED type CDATA #REQUIRED number CDATA #IMPLIED --> </pre> <p>(a)</p>	<pre> <video alias="movie-trailer"> <video alias="movie-trailer"> <instance alias="1st" description="first version" mime-type="video/mpeg" size-x="320" size-y="256" bit-rate="1.5" compression-type="mpeg2" resolution="150" duration="3.5"> <query IP-address="..." port="..." database-name="..."> select video from trailers where key="15" </query> </instance> </video> </pre> <p>(b)</p>
---	---

Fig. 2. XML metadescriptions

The meta-description of a relational database table comprises attributes concerning the location of the DBMS (with an IP address and a database name), the name of the table itself and an alias used to reference it. Sub-elements are the columns of the table, identified by a name (optionally a number) and a type, which can be a predefined SQL type or a complex type described by another meta-description.

As an example, if a column of an object-relational table contains objects of a complex type representing a MPEG video (see e.g. *Oracle's ORDVVideo* type [13]), its specific meta-description can have the valid structure of Figure 2b.

3 System Architecture

The overall architecture of the system comprises the run-time system and a set of the authoring tools; they have been designed and are now under implementation.

3.1 The Run-Time System

The run-time system supporting the XAHM model has a *three-tier* structure (Fig. 3), comprising the *Presentation*, the *Application* and the *Data Layers*.

The Presentation Layer receives final pages to be presented and eventually scripts or applets to be executed (to detect e.g. local time, location, available bandwidth or the time spent on pages). The terminal User Agent (browser) usually communicates the kind of user's terminal and its software details (OS, browser, etc.).

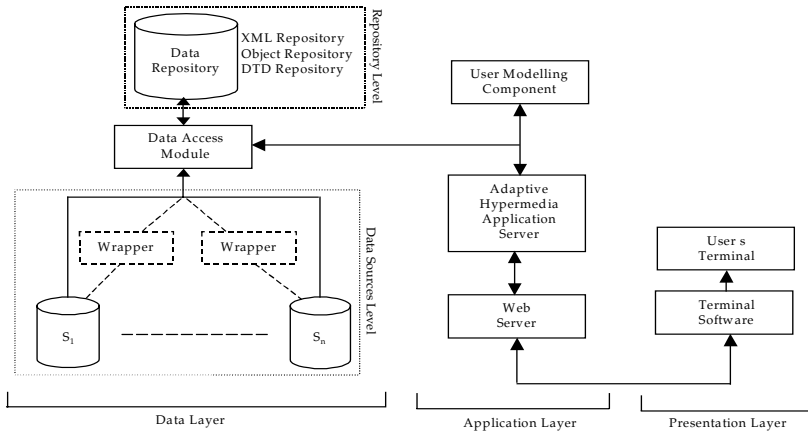


Fig. 3. Run-time system architecture

At the Application Layer there are two main modules: the *Adaptive Hypermedia Application Server (AHAS)* and the *User Modelling Component (UMC)* [4]; they run together with a *Web Server*. The UMC maintains the most recent actions of the user and executes the algorithm for the evaluation of the user's profile. The AHAS performs the Adaptation Process, executing the following steps:

1. extraction of the Presentation Description to be adapted;
2. receipt from the UMC of the user position into the adaptation space;
3. composition of the final page, composing data fragments extracted from the data sources on the basis of the user position;
4. transformation of the final page into the target (terminal) language using a terminal-dependent XSL style sheet.

Finally, the Data Layer stores persistent data and offers efficient access primitives. Each of the data sources S_i is accessed by a *Wrapper* software component that also generates, in a semi-automatic way, the XML metadata describing the data fragments stored in S_i . The *Data Access Module* implements an abstract interface for accessing the Data Sources and the Data Repository.

3.2 The Authoring Tools

The Authoring Tools have been designed to efficiently support the design, validation and testing of the hypermedia before its deployment in a running system. A *Author Module* allows to define in a graphic way the logical structure of the hypermedia, to browse the data sources, to access the associated metadata, to compose fragments to build the Presentation Descriptions and to validate them.

A *Hypermedia Simulator* allows simulating the behaviour of the AHS (i.e. the produced probabilistic density functions and the user profile assignments) on the basis of different kinds of users supposed to interact with the system (i.e. moving through its probabilistic structure). In particular, the author can:

1. analyse the intrinsic properties of the hypermedia calculated from its structure;
2. define a set of *User Classes* that describe the behaviour of typical users; many different *User Masks* can be assigned to each class, so the behaviour of each user can change during the same interaction with the system;
3. analyse the response of the AHS with respect to the User Classes.

4 Related Work

Some interesting Data Models have been used recently to describe and develop (Adaptive) Hypermedia Systems.

In the *Hypermedia Design Model (HDM)* [12] the main design phases are:

- *in-the-large*, where the overall hypermedia architecture and behaviour are designed;
- *in-the-small*, where the organisation of the overall hypermedia is detailed;
- *structure*, where the contents organisation is constructed;
- *dynamics*, where the dynamics of the hypermedia are designed.

The *Object-Oriented Hypermedia Design Model (OOHDM)* [14] represents the object-oriented evolution of HDM; it introduces the classical aspects of this paradigm and uses the abstraction and composition mechanisms. The main design phases are:

- *conceptual design*, where a conceptual model of the Application Domain is built;
- *navigational design*, where the navigational structure is designed;
- *abstract interface design*, where an abstract model for the object definition is constructed in terms of interface classes;
- *implementation*, where the previously-designed software objects are created and the architecture of the hypermedia (client-server, three-tier, etc.) is defined.

WebML [8] defines a notation for the specification and the design of a hypermedia on the basis of some constructs, formally defined in XML and intended for describing the application contexts at a high level of abstraction, disregarding implementation details. The design phase in WebML is carried out composing a set of orthogonal abstraction models:

- *structural model*, for the conceptual organisation of data;
- *derivation model*, for the extension of the structural model by deriving new information from the existing;
- *hypertext model*, for the description of some different hypertext views of the hypermedia;
- *composition model*, for the organisation of each of the hypertexts in pages and each page in information units;
- *navigational model*, for the description of how relationships between data can be transformed in navigational options;
- *presentation model*, for the definition of the layout of pages, in a terminal-independent way;
- *user model*, for the description of users and their organisation in groups;
- *personalization model*, for the definition of Event-Condition-Action rules for the hypermedia personalization.

In XAHM, the design of an adaptive hypermedia comprises the following phases:

- *logic design*, where the logic structure of the hypermedia is designed at different abstraction levels;
- *data model design*, where basic data fragments and the Presentation Descriptions are parameterised on the basis of the three adaptivity dimensions, using XML metadata;
- *presentation design*, where the XSL models containing the layouts for different user terminals and software browsers are defined.

A qualitative comparison between the four models is shown in Table 1. The considered characteristics are expressiveness, modularity, extensibility (referred to the model itself), and adaptive hypermedia support.

Table 1. Hypermedia modelling language comparison

	Expressiveness	Modularity	Extensibility	Adaptivity support
HDM	low	low	low	low
OOHDM	very good	medium	medium	medium
WebML	good	very good	very good	good
XAHM	medium	good	good	very good

It is possible to note some similarities between XAHM and WebML. Both use XML as basic language; the WebML structural model corresponds to the fragments metadata definition in XAHM; the WebML hypertext and presentation models correspond to the XAHM presentation design; the WebML composition, user and personalization models correspond to the XAHM logic design. Currently, XAHM does not support data derivation as in WebML, but it is designed to take into account the behaviour of many users to dynamically redesign the overall hypermedia structure (e.g. adding or pruning profiles). Similarly to HDM, XAHM employs a hierarchic data model, but it does not support the object-oriented paradigm as in OOHDM. This explains its extension and derivation lacks.

Although WebML offers many useful characteristics to describe AHs, we chose to develop XAHM to model with a more systematic approach the adaptation process.

5 Conclusions and Future Works

In this paper we presented a XML-based modular architecture for the modelling and the run-time support of web-based Adaptive Hypermedia Systems also supporting multi-channel access. The Adaptive Hypermedia is described as a three-dimensional space along the *user's behaviour*, *technology*, and *external environment* adaptivity dimensions. So, the adaptation process is implemented finding the proper position of the user in that space, loading and adapting the corresponding XML page, applying to it the constraints bound to that point. We are currently implementing a prototype of the system, that uses *Java* and *XML* to be completely cross-platform and extensible, and is based on the Apache [3] server suite, comprising the *Apache HTTP Server*, the *Cocoon* Application Server, the *Tomcat* Java Servlet engine, the *Xerces* XML parser and the *Xalan* XSLT Processor.

The data centric approach in the data fragments representation and the use of orthogonal concepts in the modelling phase is currently used to implement terminal- and network-independent hypermedia systems. In future works we intend to use such concepts as a basis to implement *environment-aware* web-based systems, such as the *location-dependent* services of mobile systems.

References

1. Abiteboul, S., Amann, B., Cluet, S., Eyal, A., Mignet, L., Milo, T., "Active views for electronic commerce", in *Proceedings of the 25th VLDB Conference*, 1999.
2. Adaptive Hypertext and Hypermedia Group, <http://www.wis.win.tue.nl/ah/>
3. Apache Software Foundation, "The Apache XML Project", <http://xml.apache.org/>.
4. Ardissono, L., et al., "A configurable system for the construction of adaptive virtual stores", *World Wide Web Journal*, Baltzer Science Publisher, 1999.
5. Brusilovsky, P., "Methods and techniques of adaptive hypermedia", in *User Modeling and User Adapted Interaction*, v.6, n.2-3, 1996.
6. Cannataro, M., Pugliese, A., "An XML-based architecture for adaptive web hypermedia systems using a probabilistic user model", in *Proceedings of the IEEE IDEAS Conference*. IEEE Computer Society Press, 2000.
7. Cannataro, M., Cuzzocrea, A., Pugliese, A., "A probabilistic approach to model adaptive hypermedia systems", in *Proceedings of the Int. Workshop for Web Dynamics*, 2001.
8. Ceri, S., Fraternali, P., Bongio, A., "Web Modelling Language (WebML): a modelling language for designing web sites", WWW9 Conference, 2000.
9. P. De Bra, P., Brusilovsky, P., Houben, G.J., "Adaptive Hypermedia: from systems to framework", *ACM Comp. Surveys, Symposium Issue on Hypertext and Hypermedia*, 1999.
10. De Bra, P., "Design issues in adaptive web-site development", in *Proceedings of the second workshop on adaptive systems and user modeling on the WWW*, 1999.
11. Diestel, R., *Graph Theory*, Springer-Verlag, New York, 2000.
12. Garzotto, F., Paolini, D., Schwabe, D., "HDM - A model-based approach to hypermedia application design", *ACM Transactions on Information Systems*, Jan., 1993.
13. Oracle Corporation, "Oracle *interMedia* Audio, Image, and Video Reference", Documentation Library, 2000.
14. Schwabe, D., Rossi, G., "The Object-Oriented Hypermedia Design Model", *CACM* 38(8): 45-46, 1995.
15. World Wide Web Consortium, "Extensible Markup Language", Recommendation, 2000.

Modelling the ICE Standard with a Formal Language for Information Commerce^{*}

Andreas Wombacher¹ and Karl Aberer²

¹ GMD-IPSI, Integrated Publication and Information Systems Institute, 64293 Darmstadt, Germany, wombach@darmstadt.gmd.de

² EPFL, Swiss Federal Institute of Technology, 1015 Lausanne, Switzerland, karl.aberer@epfl.ch

Abstract. Automatizing information commerce requires languages to represent the typical information commerce processes. Existing languages and standards cover either only very specific types of business models or are too general to capture in a concise way the specific properties of information commerce processes. We introduce a language that is specifically designed for information commerce. It can be directly used for the implementation of the processes and communication required in information commerce. We demonstrate the use of the language by applying it to an important standard for specifying information commerce processes, the ICE Information and Content Exchange protocol [ICE1]. By doing so we also illustrate the benefit of using formal specifications for information commerce processes allowing to capture informal specifications, like ICE, in a concise way.

1 Introduction

As modern markets move rapidly onto electronic platforms, ecommerce and ebusiness are becoming key terms in today's economy. Ecommerce addresses the trading of physical goods, such as books, food, computers and appliances. Information commerce, i.e. trading information goods, like news, software, or reports, is even more attractive over electronic channels, since goods can be distributed through the same infrastructure. We find nowadays many popular examples of information commerce on the Internet. This ranges from commercial services originating in the old economy, like digital libraries provided by scientific publishers, over new economy applications, like auction market places or information portals, to information exchange communities, like Napster or Gnutella. The business models underlying these information commerce applications are numerous and complex.

We envisage an infrastructure for information vendors who sell specific pieces of information, information mediators, who buy, recombine and resell information, and information brokers who provide directories of information vendors

^{*} This work has been partially supported by the European Commission under contract IST-1999-10288 (OPELIX) as part of the Information Technology Society program.

together with added-value information. We assume that information has an associated value, that requires controlled access in an individualized manner and guarantees concerning the quality and authenticity of the information. In such a setting the different properties associated with an information product and the corresponding interaction between buyers and sellers need to be specified in a highly configurable business process language. This is an adequate assumption for many application types, like portal sites, electronic news services, stock market information services, software evaluation services, or directory services.

In [AW01] we introduced a business process language that has been specifically developed to model information commerce scenarios. The language allows to compose processes in a flexible way by means of condition-action-rules. In order to model contractual constraints, we introduced a concept of obligations that can be derived from process specification. This allows to specify which actions are obligatory to be executed in a given process state. We have shown already, that our approach is sufficient to model simple business processes.

Within this paper, we apply this business process language to model the comparably complex ICE protocol. The Information and Content Exchange (ICE) protocol [ICE1, ICE2] is a standardized model to describe the exchange of content, like catalog data or news items, in closed user group B2B applications. It provides a negotiation protocol to determine the delivery modalities and an exchange protocol to perform the information exchange itself. In this respect the ICE protocol allows to capture some important concepts of an information commerce system, namely the negotiation and flexible specification of information exchange processes and their execution.

Our motivation to model ICE within a formal business process language is twofold. On the one hand, it serves for us as a test case for the applicability of our language. Being able to represent a complex standard specification in our formal language demonstrates its expressibility and generality. On the other hand, providing a formal specification for ICE is also useful in itself. In particular, we explicate the definition of the processes and execution constraints that are given in the ICE standard specification only informally. In that way we can step forward to a more formal specification of the standard, provide a more concise and compact description of the ICE processes and lay the foundation for the eventual verification of process properties. For example, we will model all the informal obligations that are committed in the ICE negotiation protocol, and have to be satisfied in the ICE information exchange protocol. In addition, from the specification the required message types and possible message exchanges of the ICE protocol can be derived.

In the following Section 2 we briefly mention the business process language for information commerce, that we originally proposed in [AW01] and that we use to model the ICE protocol. Section 3 is the key section and contains a brief overview of the ICE protocol and the description of the formal model for ICE. In Section 4 we give an analysis of related approaches to modelling electronic commerce processes. We conclude with a description of future work in section 5.

2 Description of the Business Process Language

We give a short introduction on the business process language that we will use in the following by means of an example. More details and motivation can be found in [AW01]. The example describes the registration of a customer requesting information from a information vendors web site. First the roles of the involved parties are listed: customer and vendor. Then the exchanged goods are specified: registration and base_info. The goods are parametrized, i.e. by assigning an url to base_info or name and email address to the registration. For each good the provider (owner) and receiver (user) is specified. For each good three standard actions can be performed: request by the user, promise and deliver by the owner. Request and promise are used to agree on the product parameters. As soon as a product is requested and promised with the same parameters and *obligation* on the side of the owner occurs to perform the deliver action, with all parameters bound to concrete values.

The performance of the actions can be further constrained by rules which establish a business workflow. The rule conditions can refer to actions performed on specific goods and lead to an explicit state change (reflected by predicates requested, promised, delivered). For example, one rule expresses that the user is allowed to request the base_info information after he has delivered the registration information. In addition, implicit state changes that are not the direct consequence of executing one of the actions, i.e. that occur without exchanging a message between the trading partners, can be specified by substitution rules. For example, the substitution rule expresses that if a registration is received by the vendor it automatically implies a promise from his side to deliver the information. In case this information has been requested earlier an obligation occurs. This is a way to express conditioned obligations.

roles: customer, vendor;

goods: registration(name: STRING, email:STRING): cust \rightarrow vend;
base_info(url:STRING): vend \rightarrow cust;

rules:

\rightarrow deliver(cust, vend, registration(n, e));

delivered(cust, vend, registration(n,e)) \rightarrow request(cust, vend, base_info(url));

...

substitutions:

delivered(cust, vend, registration(n,e),t) \Rightarrow promised(vend, cust, base_info(url));

...

In the following we will use the ICE protocol in terms of the business process language.

3 Modelling of the ICE Protocol

The ICE protocol [ICE1, ICE2] for information and content exchange is a protocol for the communication among content providers (called syndicators) and their users (called subscribers). The ICE protocol is a closed user group protocol,

which defines the roles and responsibilities of the participants in the information exchange, the syndicators and subscribers, the message formats and the method of content exchange, but leaves other dimensions like payment, authentication or metadata for content description open as orthogonal dimensions. The three main phases in the execution of the protocol are

- establishment of the ICE infrastructure
- subscription establishment and management
- data delivery.

The first phase is out of the scope of the protocol itself, but required to set up legal conditions (like copyright conditions and payment) and IT infrastructure (like access conditions) to be able to process the protocol. Based on this contract, the local ICE infrastructures of the syndicator and the subscriber must be configured. After this initial phase, the second phase starts with an interaction between syndicator and subscriber for establishing a subscription. Typically the subscriber first obtains a catalog of offers from the syndicator and then subscribes to a particular offer by proposing it to the syndicator. Alternatively the two parties may engage in a parameter negotiation protocol. After the subscriber subscribes to a particular offer the third phase, the data delivery phase, starts. ICE uses a sequenced package concept. Packages encapsulate contents of arbitrary type. ICE also defines push and pull data transfer models and detailed temporal and quantitative constraints on delivery of packages.

A subscription negotiated in the ICE protocol contains a description of temporal and quantitative constraints for performing the delivery. In particular, the subscription specifies the process of the delivery by using a constraint-based approach. Thus, ICE subscriptions can be seen as an approach to flexibly specifying business processes for content delivery. In the following we will model the ICE specification in terms of the business language mentioned in Section 2, in order to demonstrate the expressiveness of our language and to formalize the semantics of the ICE specification.

3.1 Analysing the Expressive Power of ICE Subscriptions

As mentioned above, a subscription contains the process specification for the content delivery phase within the ICE protocol. In order to model this in terms of the business process language, we first analyse ICE subscriptions in detail. To do so, we have to look at the `ice-delivery-rule` element, since it contains the constraints with respect to the delivery process. The type definition in XML (DTD) syntax is given below [ICE1].

```
<!ELEMENT ice-delivery-rule (ice-negotiable*) >
<!ATTLIST ice-delivery-rule
    mode          (push | pull)      #REQUIRED
    monthday      NMTOKENS           #IMPLIED
    weekday       NMTOKENS           #IMPLIED
    startdate     CDATA               #IMPLIED
    stopdate      CDATA               #IMPLIED
```

starttime	CDATA	#IMPLIED
duration	CDATA	#IMPLIED
minfreq	CDATA	#IMPLIED
maxfreq	CDATA	#IMPLIED
mincount	CDATA	#IMPLIED
maxcount	CDATA	#IMPLIED
url	CDATA	#IMPLIED >

We briefly describe the semantics of the different attributes. A detailed description can be found in [ICE1]. The mode attribute is the only required attribute of the ice-delivery-rule element and contains one of the values ‘push’ and ‘pull’ representing the delivery type. The attributes monthday and weekday restrict the delivery to a specific set of days within a month or a week. The startdate and stopdate attributes represent the earliest and latest point of time for content delivery. The attribute startdate can be extended by using the starttime attribute. The duration attribute specifies the period of time beginning at start-time during which delivery actions can be performed. In addition to this the minfreq and maxfreq attributes can be used to specify the minimal and maximal amount of time between two content delivery actions. All time parameters are given with seconds as the finest granularity. In addition to the temporal constraints, the mincount and maxcount attributes are quantitative constraints limiting the total number of content delivery actions.

The url attribute specifies the address where to send the update, in case it is not sent to the normal ICE communication endpoint. This allows package delivery to a different location than the other ICE communication. This is not affecting the business process itself and therefore will not be further considered here.

In order to model delivery rules in the business process language we first define a complex data type DEL_RULE that includes the parameters of the ice-delivery-rule element.

DEL_RULE: mode: {‘push’, ‘pull’},	monthday: INT,
weekday: INT,	startdate: DATE, stopdate: DATE,
starttime: TIME,	duration: TIME, minfreq: TIME,
maxfreq: TIME,	mincount: INT, maxcount: INT;

We define some functions which are convenient to access the different components of a value of type DEL_RULE. The functions startdate(d) and stopdate(d) extract the startdate and stopdate parameter from the delivery rule value d. The functions mode(d), monthday(d), weekday(d), starttime(d), duration(d), minfreq(d), maxfreq(d), mincount(d) and maxcount(d) are defined similarly. The defined functions return NULL if the parameter is undefined. In order to express temporal conditions we introduce some functions for the domains DATE and TIME. If t is a time value then DAY_OF_MONTH(t) gives the current day of the month and DAY_OF_WEEK(t) gives the current weekday. DATE(t) gives the date at time t and DAYTIME(t) gives the time of the day at time t. The expression NOW gives the current time. The parameter id is the ice-package counter and the parameter last.delivery the time of delivery of the last package. Now we can formulate a predicate to evaluate whether a package is to be delivered at time NOW as follows.

```

SAT(d, id, NOW, last_delivery) =
  (monthday(d)=NULL or DAY_OF_MONTH(NOW) in monthday(d)) and
  (weekday(d)=NULL or DAY_OF_WEEK(NOW) in weekday(d)) and
  DATE(NOW) ≥ startdate(d) and
  (stopdate(d)=NULL or DATE(NOW)<stopdate(d)) and
  (starttime(d)=NULL or DATE(NOW)>startdate(d) or
  (DATE(NOW)=startdate(d) and DAYTIME(NOW)≥starttime(d))) and
  (duration(d)=NULL or ((starttime(d)=NULL and
  (DATE(NOW)-startdate(d))*24h+DAYTIME(NOW)<duration(d))) or
  (starttime(d)≠NULL and
  (DATE(NOW)-startdate(d))*24h+DAYTIME(NOW)<duration(d)+starttime(d))
  and (minfreq(d)=NULL or NOW-last_delivery>minfreq(d)) and
  (maxfreq(d)=NULL or NOW-last_delivery<maxfreq(d)) and
  (maxcount(d)=NULL or id≤maxcount(d))

```

The start time must be always defined and is therefore not tested for NULL. There exist parameter settings, which lead to unsatisfiable constraints. Those must be detected by the syndicator and he should take care that they are not subscribed.

3.2 Modelling the Information Goods

Now we model the information items that are exchanged in the ICE protocol. We define the two participating roles (subscriber and syndicator). Then we can identify five types of information that are exchanged. Initially a catalog is sent from the syndicator to the subscriber. The catalog consists of offer elements that are selected by the subscriber and sent to the syndicator. In turn a syndicator may confirm an offer and make it a subscription. After that he sends ice-package elements with the actual content to deliver which is represented by an abstract type CONTENT.

roles: Syndicator, Subscriber;

goods:

```

catalog(offers:DEL_RULE): Syndicator → Subscriber;
offer(del_rule: DEL_RULE): Syndicator → Subscriber;
offer(del_rule: DEL_RULE): Subscriber → Syndicator;
subscription( subscription-id: ID, d: DEL_RULE): Syndicator → Subscriber
ice-package( package-id: ID, subscription-id: ID, old-state: STRING,
  new-state: STRING, cont: CONTENT): Syndicator → Subscriber

```

A few interesting observations can be made at this point. Though the same information is delivered, e.g. with an offer and a subscription, the semantics of this delivery is a different one. In addition the different information items reference each other through identifiers. The ice-package references both the subscription it is based on as well as the other ice-package which it follows.

3.3 Modelling the Delivery Phase

Based on these definitions of the goods, we are now modelling push and pull delivery as defined in the ICE protocol. The basic rule for delivering a package

is the following. Whenever an ice-package corresponding to a subscription has been promised by the syndicator and requested by the subscriber it has to be delivered. In addition the previous package in the package sequence must have been also delivered. This is expressed by the following rule

rules:

- (1) delivered(Syndicator, Subscriber, subscription(sid,d)) and
delivered(Syndicator, Subscriber, ice-package(pid,sid,old1,old,c)), last_delivery)
and promised(Syndicator, Subscriber, ice-package(pid,sid,old,new,c)) and
requested(Subscriber,Syndicator, ice-package(pid,sid,old,new,c)) and
SAT(d, sid, NOW, last_delivery)
→ deliver(Syndicator, Subscriber, ice-package(pid,sid,old,new,c))

For the first package we need a special rule since no previous packages have been delivered

rules:

- (2) delivered(Syndicator, Subscriber, subscription(sid-1,d), last_delivery) and
promised(Syndicator,Subscriber, ice-package(pid,sid,'ice_initial',new,c)) and
requested(Subscriber,Syndicator,ice-package(pid,sid,'ice_initial',new,c))
and SAT(d, sid, NOW, last_delivery)
→ deliver(Syndicator, Subscriber, ice-package(pid,sid,'ice_initial',new,c))

The time last_delivery of delivering the subscription is the reference time used to evaluate the SAT predicate. Next we give the rules that lead to the promises of the syndicator for delivering the ice-packages. These are a consequence of the subscription message, thus no more message exchanges are performed to establish those promises. Therefore we model this as an implicit state transition by a substitution rule. The promises are given stepwise with each package for the next one. Promising packages one after the other just in time simplifies the handling of cancellations and changes of subscriptions.

substitutions:

- (3) delivered(Syndicator, Subscriber, subscription(sid,d),t) and $t \leq \text{starttime}(d)$
⇒ promised(Syndicator, Subscriber, ice-package(1,sid,'ice-initial',new,c))
- (4) delivered(Syndicator, Subscriber, subscription(sid,d)) and
delivered(Syndicator, Subscriber, ice-package(pid,sid,old,new,c))
⇒ promised(Syndicator, Subscriber, ice-package(pid+1,sid,new,new',c'))

The difference among push and pull delivery lies in the way of how commitments arise from request issued by the subscriber. In case of a push delivery there exists an implicit agreement that the subscriber accepts any subscription that he receives from the syndicator. We model this by substitution rules such that the sending of the subscription message automatically implies the subscriber's requests for the first ice-package to be delivered by the syndicator. In the same way as the promises of the syndicator are established, every time package is requested by the subscriber when the previous one is delivered. This happens without the subscriber sending any requests to the syndicator and gives therefore rise to the following substitution rules.

substitutions:

- (5) delivered(Syndicator, Subscriber, subscription(sid,d),t)
and mode(d)='push' and $t \leq \text{starttime}(d)$
⇒ requested(Subscriber, Syndicator, ice-package(1,sid,'ice-initial',new,c))
- (6) delivered(Syndicator, Subscriber, subscription(sid,d)) and mode(d)='push' and
delivered(Syndicator, Subscriber, ice-package(pid,sid,old,new,c))
⇒ requested(Subscriber, Syndicator, ice-package(pid+1,sid,new,new',c'))

Together with the promises of the syndicator these rules imply that the syndicator is obliged to deliver the packages. This completely specifies the mechanism of push delivery.

For pull delivery request messages are sent by the subscriber for each package. The only limitation on making of requests in the ICE standard is the requirement that a subscription for pull delivery exists. The responsibility for making the correct requests remains with the subscriber. This is expressed by the following rule for executing requests by the subscriber.

rules:

- (7) $\text{delivered}(\text{Syndicator}, \text{Subscriber}, \text{subscription}(\text{sid}, \text{d})) \text{ and } \text{mode}(\text{d}) = \text{'pull'}$
 $\rightarrow \text{request}(\text{Subscriber}, \text{Syndicator}, \text{ice-package}(\text{pid}, \text{sid}, \text{old}, \text{new}, \text{c}))$

As the syndicator promises automatically the delivery of the packages, each time such a request is issued and the conditions of the first two rules on delivery turn true, the delivery becomes obligatory for the syndicator.

3.4 Subscription Establishment and Management

The subscription establishment and management phase can be partitioned into different activities

- catalog handling
- negotiation resulting in either end of communication or delivery of a subscription
- renegotiation of already existing subscriptions and potentially changing it
- cancellation of an existing subscription

Within this paper, we treat only the first two points because of space limitations. We start with the catalog handling. In the ICE protocol it is specified, that the subscriber can request a catalog, which has then to be delivered by the syndicator whereby this catalog contains all offers provided by the syndicator. It is important to mention, that no obligations are implied by sending these offers. The syndicator is able to retreat any offer unless he has sent a subscription. The rules related to sending the catalogues are straightforward.

rules:

- (8) $\rightarrow \text{request}(\text{Subscriber}, \text{Syndicator}, \text{catalog}(\text{o}))$
 (9) $\text{requested}(\text{Subscriber}, \text{Syndicator}, \text{catalog}(\text{o}))$
 $\rightarrow \text{deliver}(\text{Syndicator}, \text{Subscriber}, \text{catalog}(\text{o}))$

The ice-get-catalog message does not require any precondition and can be performed anytime. The catalog is delivered after it has been requested, though the syndicator is not obliged to do so. The ICE specification does not state whether the syndicator must deliver the catalog or may deliver it. In case the delivery is obligatory this can be modelled by adding an init clause where the catalog delivery is promised by the syndicator. The negotiation model of the ICE protocol is represented by the following state diagram.

The diagram can be divided into two parts. The left side is associated with the syndicator, while the right side is associated with the subscriber. The grey circle represents the starting state and the annotations of the arcs depict the

message types. The subscriber starts with an offer, which can be accepted by the syndicator resulting in a subscription (final state of negotiation). The syndicator can also respond with a counter offer or can reject the request resulting in a final state. If the syndicator sends a counter offer, the subscriber has similar possibilities. The subscriber now can either send the unchanged offer back to the syndicator to indicate acceptance, response with a further counter offer or reject the offer, resulting in a final state.

It is important to mention, that within this negotiation model no obligations are specified in the ICE protocol, although they might be useful.

The rules for modelling this negotiation model are given below.

rules:

- (10) $\rightarrow \text{deliver}(\text{Subscriber}, \text{Syndicator}, \text{offer}(d))$
- (11) $\text{delivered}(\text{Subscriber}, \text{Syndicator}, \text{offer}(d))$ and $d \neq d'$
 $\rightarrow \text{deliver}(\text{Syndicator}, \text{Subscriber}, \text{offer}(d'))$
- (12) $\text{requested}(\text{Subscriber}, \text{Syndicator}, \text{subscription}(\text{sid}, d))$
and $(\text{starttime}(d) \neq \text{NULL} \text{ or } \text{starttime}(d) = \text{NOW})$
 $\rightarrow \text{deliver}(\text{Syndicator}, \text{Subscriber}, \text{subscription}(\text{sid}, d))$

substitution:

- (13) $\text{delivered}(\text{Subscriber}, \text{Syndicator}, \text{offer}(d))$
 $\Rightarrow \text{requested}(\text{Subscriber}, \text{Syndicator}, \text{subscription}(\text{sid}, d))$

The first rule specifies that offers can be sent by the subscriber, without any former action (as described in section 4.3.2 in [ICE1]). The second rule models the counter offer sent by the syndicator which requires the sending of a previous offer by the subscriber. The fact that the counteroffer must be different is not clearly stated in the ICE specification, but is implied by comments in the specification. The third rule states that only requested subscriptions may be delivered. A subscription is implicitly requested if and only if the subscriber has proposed the corresponding offer, by sending it to the syndicator. This relationship is modelled by the substitution rule. Rejection of offers and subsequent termination of the negotiation needs not to be modelled, as it is simply realized by not continuing the communication.

This model also shows that the constraints for negotiation under the ICE protocol are extremely weak. All the negotiation steps are indicative for the other side, but imply no obligations. For example, offers from the catalog are not binding, offers can be created completely independent of the ones contained in the catalog, and offers sent during negotiation are not binding as well. The only execution constraint is that only those subscriptions are possible that have been explicitly proposed by the subscriber. Obligations, as we have seen, occur however, once a subscription is made. Then the packages have to be delivered according to the specification of the subscription.

3.5 Execution Constraints of the ICE Protocol

One of the goals of modelling the ICE protocol with our business process language is to give a formal specification of the constraints on the execution of the ICE protocol. These are given in the ICE specification in an informal manner.

This shows that it is feasible to express also a complex standard specification within a formal framework. As a result the specification is given in a more compact form and can provide implementors of the standard with an unambiguous specification. In the table below we list some of the important execution constraints of the ICE protocol. We identify the section where the corresponding specification can be found in the ICE specification document [ICE1], cite the textual description of the ICE specification and identify the rules of our formal specification that capture that constraint. The relationship between constraints and rules is of course not one-to-one. For example, a specification like 'ICE uses a sequenced package model' results in several rules that are needed to express the corresponding semantics formally, i.e. rules (4) and (6). See table for a detailed description of mappings.

We also would like to point out that our specification of the ICE protocol is not complete. The most important part of the specification we left out is the renegotiation of subscriptions, which currently can not be modelled with our approach in a straightforward manner. The corresponding messages of the ICE protocol are ice-change-subscription or ice-cancel. Further, we didn't explicitly modelled the uniqueness of the subscription id (sid) within rule 12 and we assumed, that only one package is transmitted within each content delivery action while the ICE specification allows multiple packages to be included in one delivery operation. We used this approach to reduce the complexity of the handling of the package ids without limiting the functionality of the model.

To summarize, we described in our model the main parts of the ICE protocol. We were in particular treating the negotiation of the delivery processes and the execution of the resulting delivery specification. We showed, that the business process language provides the capabilities to formally describe the semantics of such flexible process specifications, and in particular the execution constraints associated with the process that are given in the ICE specification in an informal manner.

4 Related Work

Mechanisms for the informal and formal specification of interaction processes in electronic commerce are described in many different contexts, both in standardization and research. In the following, we give a brief overview of approaches more or less related to the business process Language. In the context of **Web standardization** the Micropayment Markup language [MICRO] and the Information and Content Exchange (ICE) Protocol [ICE1, ICE2] are the ones closest to our work describing information commerce, but less general in defining possible business processes. Many **secure protocols** have been proposed for electronic commerce and information commerce that guarantee secure and fair exchanges. They consider electronic delivery, like Netbill [NETBILL, TYGAR99] or DigiBox [INTER, C96], or right management and granting fair exchange of goods [ASW98, GWW01]. These approaches are in general too restricted in order to allow the modelling of information commerce processes in general. **Agent communication languages**, like KQML [FFMM98] focus on the problem of

ICE - Rule Mapping

Rule number	Section in [ICE1]	Textual description in the ICE specification
2, 3	5.1.4	When it (the subscriber) first starts a new subscription, the Subscriber starts in state ICE-INITIAL.
5, 6	5.4	If a subscription has a delivery policy method of type push, the Syndicator must initiate the delivery of the packages ... containing one or more ice-package elements. When a Syndicator sends a package to the Subscriber, the Syndicator MUST provide the expected state of the subscription before and after the package is processed.
4, 6	5.1.2	ICE forces a Syndicator (and a Subscriber) to view the package stream as a strictly ordered sequence of packages. This means that packages cannot be processed out of order, and all intermediate packages must be processed.
7	5.3	If a subscription has a delivery policy method of type pull, the Subscriber must initiate the delivery of the packages with the ice-get-package request.
1	5.3	When a Subscriber requests a package from the Syndicator, the Subscriber MUST provide the state of the subscription and subscription identifier,...
8	4.3.2	Subscribers can use ice-get-catalog to obtain the list of subscription offers for which they are eligible.
9	4.3.2	Return response (to an ice-get-catalog) is an ice-catalog.
10, 13	4.4, 4.5.2	A Subscriber uses the ice-offer request to establish a subscription. Typically, a Subscriber will use ice-get-catalog to get a catalog, take one of the ice-offer structures from that catalog, and send it back to the Syndicator in a request. However, the Subscriber is free to create an ice-offer structure in any implementation-defined manner it wants. For example, a Syndicator might e-mail an ice-offer to a Subscriber, who could then feed it into their ICE tool and begin the protocol processing here. AND: Negotiation begins with the Subscriber making an ice-offer request to the Syndicator.
11	4.5.2	The Syndicator indicates a counter-proposal by rejecting the ice-offer ..., and including a counter ice-offer in the response.
12	4.5.2	The Syndicator accepts the offer ... including an ice-subscription response.
10	4.5.3	If the Subscriber receives a counter proposal (that is another offer instead of a subscription), the Subscriber MAY try another ice-offer, either with the contents of the counter proposal received from the Syndicator, or with some other mixture of parameters. The method of choosing what parameters to alter is a quality of implementation issue. AND: If the Subscriber receives a Sorry response (that is no further action performed, neither a subscription nor a counter offer), the Subscriber MAY try again with some other ice-offer, although the Syndicator has (unhelpfully) not given any clues as to what to try.
13	4.4	A Subscriber uses the ice-offer request to establish a subscription.

communication among autonomous software agents. These languages correlate the agents internal states with the messages that are intended to change these states and specify the interaction languages and protocols that can be used to establish agent communication. There exist approaches to use agent communication languages in order to model electronic commerce business processes, like FLBC [KIMB, WH98].

5 Future Work

The next steps are to provide for the business process language a formal semantics in terms of dynamic deontic logic and to complete the implementation of the system, which is based on the architecture described in [WKA01] and focuses the support of a light-weight infrastructure.

References

- [ASW98] N. Asokan, V. Shoup, M. Waidner: Asynchronous Protocols for Optimistic Fair Exchange, Proc. of S&P 98, Oakland, California, 1998.
- [AW01] K. Aberer, A. Wombacher: A language for information commerce processes, Technical Report EPFL, 2001.
- [C96] B. Cox: Superdistribution, Addison-Wesley, 1996.
- [FFMM98] T. W. Finin, R. Fritzson, D. McKay, R. McEntire: KQML As An Agent Communication Language. CIKM 1994: 456-463, 1994.
- [GWW01] C. Günther, S. Weeks, A. Wright: Models and Languages for Digital Rights, Proceedings of the 34 th HICSS, 2001.
- [KIMB] S. Kimbrough: Formal Language for Business Communication (FLBC): Sketch of a Basic Theory, forthcoming in International Journal of Electronic Commerce
- [TYGAR99] J. D. Tygar: Atomcity versus Anonymity: Distributed Transactions for Electronic Commerce. VLDB 1998: 1-12, 1998.
- [WH98] H. Weigand, W.J. van de Heuvel: Meta-patterns for Electronic Commerce based on FLBC. Proc. HICSS'98, IEEE Press, 1998.
- [WKA01] A. Wombacher, P. Kostaki, K. Aberer, WebXIce: An Infrastructure for Information Commerce on the Web, Proceedings of the 34 th HICSS, 2001.

Web References

- [ICE1] W3C note of the ICE protocol.
www.w3.org/TR/1998/NOTE-ice-19981026
- [ICE2] Information and Content Exchange Protocol home page.
www.icestandard.org/
- [INTER] Intertrust home page. www.intertrust.com
- [MICRO] Micropayment Markup Language home page. www.w3c.org/ecommerce
- [NETBILL] NetBill home page. www.netbill.com
- [OPELIX] OPELIX home page. www.opelix.org

Managing Web Data through Views^{*}

Álisson R. Arantes¹, Alberto H. F. Laender¹,
Paulo B. Golgher^{1,2}, and Altigran S. da Silva^{1**}

¹Computer Science Department
Federal University of Minas Gerais
31270-901 Belo Horizonte MG Brazil
{alissonr, laender, golgher, alti}@dcc.ufmg.br

²Akwan Information Technologies
Av. Antônio Abraão Caram 430
31275-000 Belo Horizonte MG Brazil
golgher@akwan.com.br

Abstract. The huge amount of data available on the Web creates a great demand for methods and tools that allow the manipulation of such data. Thus, the notion of *view* as a mechanism for providing access to Web data has been revisited. In this paper, we present an environment composed of a set of high-level tools that allow the fetching, extraction, integration, and refreshing of Web data. Using this environment, database designers can build and maintain Web views by defining schemas for data integration, specifying wrappers (agents for collecting Web pages and extracting data from them), and defining plans for refreshing the view contents.

1 Introduction

One of the key issues in modern Web based information systems is their capability of using data extracted from different Web sites, therefore taking advantage of the huge volume of data made available by the popularization of the Web. As a result, there is an increasing need for flexible high-level tools that allow the fetching and extraction of such data. Further, there is also a need for tools that provide means to specify how these data should be integrated and maintained.

In this paper, we present an environment composed of a set of high-level tools that allow the fetching and extraction of Web data, and the definition of a plan for their integration and maintenance. We call this task *Web view definition*, since the result is a database that can be considered as a view of the data available in the original Web sites.

According to Gupta et al. [8], there are basically two approaches for generating views from Web data sources: the *virtual* approach, where data from different

^{*} This work was partially supported by Project SIAM (MCT/CNPq/PRONEX grant number 76.97.1016.00) and by CNPq (grant number 467775/00-1).

^{**} On leave from the University of Amazonas, Brazil.

Web sites are fetched and extracted *on-the-fly* during the processing of a query, and the *materialization* approach, where data are first fetched and stored in a suitable format for later querying. Although the environment proposed here can be used with either approaches, some of its features (e.g., the policies for refreshing) are more relevant when the materialization approach is considered. Therefore, in this paper, we adopt the materialization approach in our examples and discussion.

The paper is organized as follows. Section 2 discusses related work. Section 3 presents our proposed environment and introduces our tool for Web view definition, the WebView tool. Section 4 describes the WebView tool in more details. Section 5 describes the execution of the Web view materialization plan, that is the output of the WebView tool. Finally, Section 6 concludes the paper and presents future work.

2 Related Work

The work presented here falls in the category of tools for helping in the tasks of modeling, extracting, and integrating data available on the so-called *data rich* Web sources [5], what is currently being termed as *Web Data Management* [1]. In the recent literature, many systems and environments for Web data management have been proposed, some of which are discussed in this section.

The goal of the TSIMMIS [6] project is to develop tools for the integration of heterogeneous data sources, such as Web data sources. The user can construct mediators that work as an intermediate layer between the client applications and the heterogeneous data sources. Data from the data sources are fetched and integrated during the query execution, not supporting the materialization approach. Moreover, the TSIMMIS tools require manual code writing for locating and extracting the data of interest. The ARIADNE [2] system provides tools that allow the construction of mediators for heterogeneous Web data sources. These mediators can then be used for answering queries against the available data in those data sources. ARIADNE also does not support materialized views but provides some degree of automation, not requiring the user to write code for extracting data from the Web data sources. The ARANEUS [10] system also provides tools for mediator construction. Like our approach, it supports materialized views. However, in ARANEUS, the user has to manually code data extractors and view definitions over the Web data sources. The WHOWEDA (*Warehouse of Web Data*) [3] system main goal is to design and implement a Web warehouse that materializes and manages useful information from heterogeneous Web data sources. However, according to its data model, Web data are modeled having pages and links as its primary objects. Thus, the internal structure of the Web documents is not modeled, which only allows queries to be specified on the whole content of the document, regardless of its internal structure.

Our work differs significantly from the above discussed ones in the sense that it is fully based on the use of high-level tools to guide the construction and maintenance of Web views. This means that no code writing is required, therefore supporting the fast development of Web based information systems.

3 The Proposed Environment

In this section, we introduce the proposed environment for Web view definition and maintenance. This environment is composed of a set of high-level tools that allow the definition and maintenance of Web views. The environment is basically composed of three tools:

The ASByE tool - The ASByE tool generates agents for automatically collecting sets of dynamic or static Web pages. In a typical interaction with the tool, the user gives examples of how to reach the pages of interest, filling any form, if needed, and of how to group together related pages. The output of the tool is a possibly parameterized agent that fetches the selected pages [7].

The DEByE tool - The DEByE tool allows the generation of data extractors for semistructured data sources. The user gives examples of the data of interest and informs how to structure them using a nested table paradigm. The output of the tool are data extractors that generate sets of XML objects when executed [11]. The DEByE tool is specially suitable for sources presenting structural variations. For instance, it can deal nicely with missing or out of order attributes.

The WebView tool - This tool allows the definition of Web views by specifying how the data extracted from different Web sites should be integrated, materialized, and refreshed. The output of the tool is an XML file that guides the tasks of fetching and extracting the data, and materializing and maintaining the Web view. The WebView tool will be described in Section 4.

Figure 1 describes the architecture of the proposed environment. For a given set of data sources of interest $D = \{ds_1, ds_2, \dots, ds_n\}$, the user interacts with the ASByE and DEByE tools to generate a set of agents $A = \{a(ds_1), a(ds_2), \dots, a(ds_n)\}$ and a set of extractors $X = \{x(ds_1), x(ds_2), \dots, x(ds_n)\}$. Then, with the WebView tool, the user specifies how the agents and data extractors are related to each other, and how the data extracted will be integrated. Further, the user specifies how the Web view will be maintained. The view materialization plan generated guides the view processor in order to materialize and maintain the Web view.

We now introduce an example that will be used throughout the paper to illustrate the operation of our proposed environment. Suppose we need to build a view containing information on movies playing in New York City and Washington D.C. The data for populating this view will be extracted from pages taken from the Web sites of the *New York Times* (<http://www.newyorktimes.com>) and the *Washington Post* (<http://www.washingtonpost.com>) newspapers. Besides, the view will include complementary specific data on the movies (director, genre, etc.) that are not available in these sites and will be taken from the IMDb Web site (<http://www.imdb.com>). Figure 2 shows sample pages from these three Web sites.

In order to populate our view, we will have to build a data extractor, using the DEByE tool, for each one of the Web sites. For the *Washington Post* Web site, for instance, its data extractor will take as input the page shown in Figure 2 and will output the objects (tuples) illustrated in Figure 3. A set of objects with the same attributes will be generated by the extractor built for the *New York*

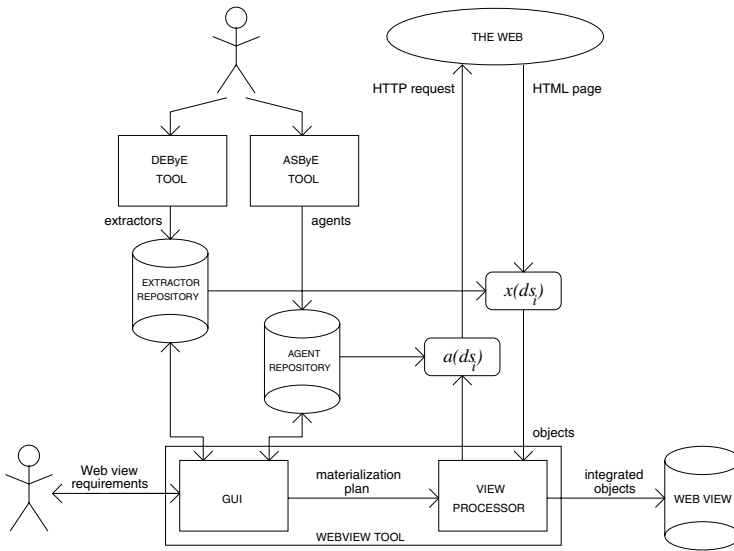


Fig. 1. Architecture of the proposed environment.

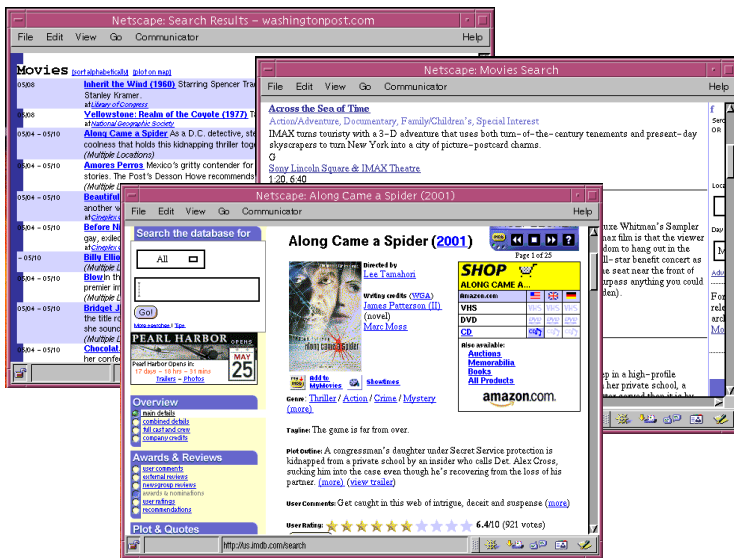


Fig. 2. Sample pages from the three Web sites used in our example.

Times Web site. As for the objects extracted from the IMDb Web site, they will have the structure of the tuples in the table shown in Figure 4.

Using standard relational model notation, we can define the sets of objects extracted from each site as being relations with the following schemes:

Title	Synopsis	Playing In
<i>Inherit the Wind (1960)</i>	<i>Starring Spencer Tracy...</i>	<i>Library of Congress</i>
<i>Yellowstone: Realm of the Coyote (1977)</i>	<i>Tales of survival in a national park.</i>	<i>National Geographic Society</i>
<i>Along Came a Spider</i>	<i>As a D.C. detective...</i>	<i>Multiple Locations</i>
<i>Amores Perros</i>	<i>Mexico's gritty contender...</i>	<i>Multiple Locations</i>
<i>Beautiful Creatures</i>	<i>Rachel Weisz stars in...</i>	<i>Cineplex Odeon Foundry</i>
...

Fig. 3. Objects extracted from the Washington Post Web site.

Title	Director	Genre	Rating
<i>Inherit the Wind (1960)</i>	<i>Stanley Kramer</i>	<i>Drama</i>	<i>8.0</i>
<i>Along Came a Spider</i>	<i>Lee Tamahori</i>	<i>Thriller/Action/Crime/Mystery</i>	<i>6.4</i>
<i>Amores Perros</i>	<i>Alejandro González Iñárritu</i>	<i>Thriller/Drama</i>	<i>8.5</i>
<i>Beautiful Creatures</i>	<i>Bill Eagles</i>	<i>Comedy/Thriller/Crime/Drama</i>	<i>6.4</i>
...

Fig. 4. Objects extracted from the IMDb Web site.

NYT(Title, Synopsis, Playing In)
 WSP(Title, Synopsis, Playing In)
 IMDb(Title, Director, Genre, Rating)

Thus, our view can be defined by the following relational algebra expression:

$$(NYT \cup WSP) \bowtie IMDb \quad (1)$$

We notice that all objects we are dealing with in our example are very regular and have a very simple structure. For example, objects from the **New York Times** and **Washington Post** Web sites have exactly the same structure (in relational terminology, we could say that they are *union-compatible*). Moreover, the join operation is a natural join over the attribute **Title**.

However, our environment is capable of dealing with much more complex situations, due to the modeling features of the DEByE tool, that allow the manipulation of semistructured objects with nesting structures subject to variations [9]. The simpler modeling we adopt here is for the sake of focusing our discussion on the issue of Web view definition and maintenance.

For fetching the appropriated pages from the Web sites used for populating the view, agents generated by the ASByE tool are used. There is a specific agent for each site. For instance, the agent for the **Washington Post** Web site automatically submits a query to the site and collects the set of sequential pages generated as a result, each page containing a list of data items on movies. The agent for the **New York Times** Web site has a similar behavior. The agent generated for the **IMDb** Web site takes as input the title of a movie, submits a query

using this title as an argument to the search service of the site, and collects the page returned, which contains specific data on the movie whose title is given. This feature will be necessary for executing the join operation defined in (II), as we shall see later.

Since the ASByE and DEByE tools were described in depth in previous published papers [7][11], here we focus our attention on the WebView tool, and how it interacts with the other tools.

4 The WebView Tool

The main purpose of the WebView tool is to provide a high-level graphical user interface to specify how the data fetched and extracted from various Web sites will be integrated and refreshed. Thus, this tool basically specifies when (and which of) the ASByE agents should run to feed the DEByE extractors, and how the XML objects generated by the extractors should be integrated.

The tool provides a graph-based interface in which the nodes are data sources and the arcs define relationships between these data sources, as illustrated by Figure 5. There are three types of nodes that represent the following types of data sources:

Primary Data Source - Data sources of this type are the main sources of data for a Web view. For instance, in an application dealing with movie programs, nodes of this type could represent the newspaper movie sections which will provide the essential data.

Dependent Data Source - This is a data source that provides complementary data for objects from another data source called its *master* data source. For instance, technical data on movies could be considered complementary data. These data are considered meaningless without their counterparts in the master data source.

Union Data Source - This represents a union of two or more data sources. Sources of this type can be used, for example, to represent all data available from several distinct newspapers.

Figure 5 shows the specification of the data sources used for the definition of the Web view in our running example using the WebView GUI. The primary data sources are the *New York Times* and *Washington Post* Web sites which are represented by the nodes labeled PS. The IMDb Web site is a dependent data source and is represented by the node labeled DS. The node labeled U is a union data source which includes data coming from both primary data sources. We notice that Figure 5 illustrates how the definition of the view expressed by (II) is accomplished using the WebView GUI.

As the view definition proceeds, for each primary data source the user must select an agent (generated by the ASByE tool) to fetch the appropriated pages, and an extractor (generated by the DEByE tool) to extract data from the fetched pages. When an extractor is selected, the user is also asked to select a list of *identifier attributes* from the list of attributes available for the objects extracted from the data source (which is embedded in the extractor definition). The identifier attributes have a semantic role equivalent to the one of primary keys in

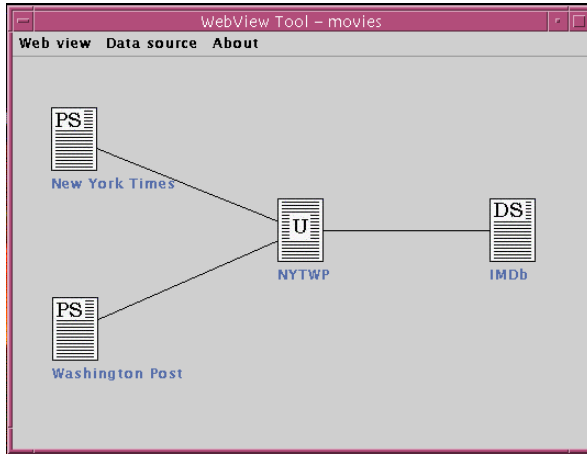


Fig. 5. A snapshot of the WebView GUI.

the relational model. In addition, the user must also define a *refreshing policy* for the primary data sources. The following policies are available:

Polling - According to this policy, the data source is constantly monitored in order to identify updates. The user defines the frequency of the polling when the data source is specified. The user can also indicate that the source will be monitored by calculating the average time in which the data in this source change.

Pushing - Here, the data source has the burden of informing the user when it has been updated. This kind of service has become a new paradigm for information delivery, also known as *Webcasting*. This is accomplished by constantly monitoring the arrival of an external file (for instance, the user mail box) specified when creating the data source node.

On demand - Besides the above policies, a Web view can also be refreshed on demand.

In Figure 6 we illustrate how the user provides the information required to specify a primary data source for the **Washington Post** Web site and a dependent data source for the **IMDb** Web site. Notice that, in this case, the attribute **Title** of the **Washington Post** Web site was the one selected as the identifier attribute.

For dependent data sources, the user must select an agent to fetch the appropriated pages and an extractor to extract the data from them, similarly to what is done for primary data sources. In addition, in the case of dependent data sources, the user is asked to provide the following information: (1) the master data source related to this dependent data source (the data in the dependent data source are considered functionally dependent on the data in the master data source) and (2) the join attributes required to perform the join operation between the data available in the dependent data source and in its related master data source.

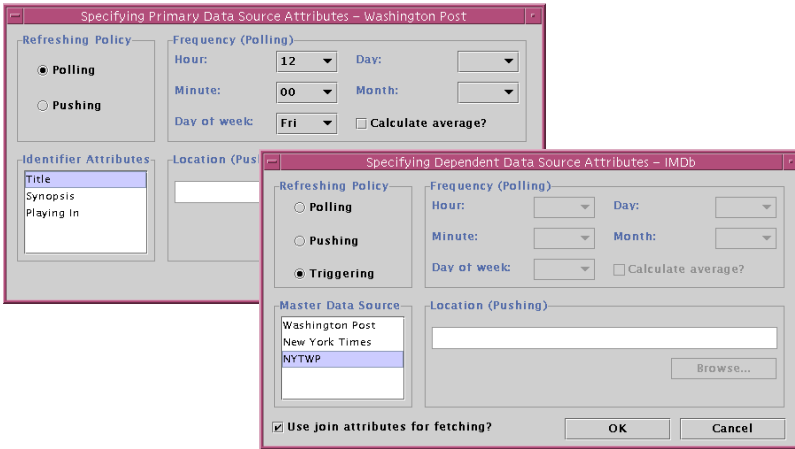


Fig. 6. Specifying primary and dependent data source attributes.

The task of identifying objects that describe the same “entity” in different sources, for performing the join between them, is one of the major difficulties when integrating data extracted from the Web. In our current implementation, two objects are considered as being the same “entity” if the attributes specified as the join attributes have exactly the same value in both sites. This naive approach certainly fails in many cases, and we plan to develop a more general approach in future versions. For instance, we can adopt the solution proposed in [4], where textual similarity is used for establishing identity between objects from distinct sources.

Concerning the refreshing policy, in the case of dependent data sources, an additional option is available. The user can specify that the data from a dependent data source must be refreshed every time the data from its master data sources are refreshed. This refreshing policy is called *triggering*, as is the case of the IMDb Web site, shown in Figure 6.

For cases where objects having the same semantics but that originate from several distinct data sources must populate a view, the user can specify a union data source. In our example a union data source was created to represent the union of sets of objects extracted from the New York Times and Washington Post Web sites (see Figure 5), what is required for the view definition in (II).

Note that the data sources composing a union data source are required to be “union-compatible”, as it is the case of the view defined by (II). In our proposed environment, this requirement can be satisfied by properly modeling the objects to be extracted from the data sources when using DEByE. We refer the interested reader to [9] for a deeper discussion of the semistructured data modeling features of DEByE.

The output of a typical interaction with the WebView GUI is an XML file that properly encodes instructions and data that will guide the view processor in the tasks of fetching and extracting data to materialize and maintain the

Web view. We call this XML file a *view materialization plan*. Figure 7 shows the resulting plan generated for the discussed example.

```
<?xml version="1.0"?>
<WEBVIEW name="movies">
  <PRIMARY identifiers="Title" name="Washington Post">
    <REFRESH policy="polling">
      <FREQUENCY day="" dayofweek="Fri" hour="12" minute="00" month="" calcavg="no"/>
    </REFRESH>
    <FETCH agent="washingtonpost.pl"/>
    <EXTRACT extractor="washingtonpost.oep.xml"/>
  </PRIMARY>
  <PRIMARY identifiers="Title" name="New York Times">
    <REFRESH policy="polling">
      <FREQUENCY day="" dayofweek="Fri" hour="12" minute="00" month="" calcavg="no"/>
    </REFRESH>
    <FETCH agent="newyorktimes.pl"/>
    <EXTRACT extractor="newyorktimes.oep.xml"/>
  </PRIMARY>
  <UNION name="NYTWP">
    <DATASOURCE name="New York Times"/>
    <DATASOURCE name="Washington Post"/>
  </UNION>
  <DEPENDENT name="IMDb" master="NYTWP">
    <REFRESH policy="triggering"/>
    <FETCH agent="imdb.pl" usejoinattributes="yes"/>
    <EXTRACT extractor="imdb.oep.xml"/>
    <JOIN>
      <ATTRIBUTE master="Title" dependent="Title"/>
    </JOIN>
  </DEPENDENT>
</WEBVIEW>
```

Fig. 7. View materialization plan generated by the WebView tool for our example.

5 Materializing and Maintaining a Web View

In this section, we describe how the view processor takes a view materialization plan as input and performs all the actions necessary for materializing and maintaining the corresponding Web view.

5.1 Materialization

When a view materialization plan is generated and first executed, the corresponding Web view must be populated with data coming from its data sources.

The first step for populating a Web view is the fetching of the pages from the primary data sources. Using the data encoded in the **PRIMARY** elements of the view materialization plan (see Figure 7), the view processor can properly invoke the necessary agents and extractors in order to produce the sets of objects of interest.

For the case of dependent data sources, only pages containing data related to the objects extracted from their corresponding master data sources must be

fetches. Once these pages are fetched, they will feed the proper extractor (previously selected by the user) resulting in a temporary data repository containing all objects extracted from the pages fetched from the dependent data source. Then, the view processor selects from the extracted objects only those satisfying the join condition previously established in the specification of the dependent data source. Thus, the final view objects are built by joining each object from the primary data sources with the related objects in the dependent data sources. These are the objects used to populate the Web view.

Concerning the fetching of pages from dependent data sources, there is a subtlety that we must observe. There are cases when the pages containing the objects to be extracted are dynamically generated as the result of filling a form with proper values. For example, the complementary information on a movie taken from the IMDb Web site can be obtained from a page that is dynamically generated as the result of submitting an HTML form containing the title of the movie to the IMDb search interface. To cope with such a situation, the agent that was generated by the ASByE tool must accept the movie title as a parameter (a feature fully supported by ASByE, as explained in [7]) and the proper value for this parameter (e.g., the movie title) must be supplied when the agent is invoked by the view processor. This is accomplished by using the join attribute provided by the user when the dependent data source was specified. In this case, the view processor will invoke the proper agent supplying the join attributes as parameters for each object obtained from the master data sources. The set of fetched pages is then used as input for the extractor. Of course, there are cases where no parameter is needed and then one invocation of the agent is enough to fetch all the needed pages.

In our example, first the objects from the New York Times and Washington Post Web sites are fetched and extracted. Next, using the information encoded in the **DEPENDENT** element of the materialization plan (see Figure 7), the view processor fetches, from the IMDb Web site, the set of pages related to the objects extracted from the master data source. In this case, one page for each master object is fetched in the dependent data source, using the join attributes (in this case, the movie title) as parameters as defined by the **usejoinattributes** attribute in the **FETCH** element of the plan. Then, using the objects extracted from the set of IMDb fetched pages, the view processor executes the join operation.

After populating the Web view, the view processor schedules its first refreshing, according to the **REFRESH** elements encoded in the plan.

5.2 Maintenance

Once materialized, a Web view must be maintained to reflect the changes in the data sources. For the primary data sources whose refreshing policy is polling, their refreshing is scheduled using the frequency parameters provided by the user. For the ones whose refreshing policy is pushing, their refreshing is accomplished by monitoring the changes of the external file used as the original data source.

During the refreshing of a Web view, the process of fetching, extracting and integrating the data from its data sources is similar to that done during its materialization. However, the main difference is that objects that are not present in

the data available on the original source have to be removed from the Web view. In order to identify the addition or removal of objects, the identifier attributes specified when the view was defined are used to compare the contents of the Web view with the current contents of the data source.

The refreshing of the Web view defined in our example is as follows. If a movie is present in the Web view and is not in the *New York Times* or *Washington Post* Web sites any more, it is removed from the Web view. On the other hand, if a movie has been extracted from the *New York Times* or *Washington Post* sites and is already in the Web view, its related page is not fetched from the IMDb Web site and therefore it is not inserted into the Web view.

Due to the fact that the refreshing process can be done at any time, without user intervention, all its details are written in a log file. With this log, the user can monitor the Web view refreshing and revise its plan if needed.

6 Conclusions

In this paper, we presented an environment that allows the definition and maintenance of views over distinct Web sources. In comparison with other approaches for view definition and maintenance, such as ARIADNE [2] and ARANEUS [10], the main contribution of our work is that it is fully based on the use of high-level tools to guide the definition of the Web views. Therefore, there is no need for code writing and the maintenance of the view definition can be done by simply revising the previous interactions with the available tools.

In addition, we notice that the only approach that presents a degree of automation closer to ours is ARIADNE. However, as our tools use high-level metaphors in their operation, our environment provides better support for fast development of Web based information systems than ARIADNE. We also believe that the materialization approach that we adopt is more convenient because the data are extracted, integrated, and stored locally. Moreover, using our refreshing policies, the changes in the original data sources will be reflected in the Web views.

For future work, we plan to develop a more general mechanism for establishing identity between two objects from distinct sources. Further, we plan to provide means for monitoring changes in the structure of the data sources, in order to promptly identify the need for revising the view definition. We also intend to investigate the issues of scalability and performance of the WebView tool.

References

1. ABITEBOUL, S., BUNEMAN, P., AND SUCIU, D. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann Publishers, Los Altos, California, 1999.

2. AMBITE, J. L., ASHISH, N., BARISH, G., KNOBLOCK, C. A., MINTON, S., MODI, P. J., MUSLEA, I., PHILPOT, A., AND TEJADA, S. ARIADNE: A System for Constructing Mediators for Internet Sources. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (Seattle, Washington, 1998), pp. 561–563.
3. BHOWMICK, S. S., NG, W. K., AND LIM, E. P. Information Coupling in Web Databases. In *Conceptual Modeling - ER '98, 17th International Conference on Conceptual Modeling*, T. W. Ling, S. Ram, and M.-L. Lee, Eds. Springer, Berlin, Germany, 1998, pp. 92–106.
4. COHEN, W. W. Integration of Heterogeneous Databases Without Common Domains Using Queries Based on Textual Similarity. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (Seattle, Washington, June 1998), pp. 201–212.
5. EMBLEY, D. W., CAMPBELL, D. M., JIANG, Y. S., LIDDLE, S. W., NG, Y. K., QUASS, D., AND SMITH, R. D. Conceptual-Model-Based Data Extraction from Multiple-Record Web Pages. *Data & Knowledge Engineering* 31, 3, 227–25.
6. GARCIA-MOLINA, H., HAMMER, J., IRELAND, K., PAPAKONSTANTINOY, Y., ULLMAN, J., AND WIDOM, J. Integrating and Accessing Heterogeneous Information Sources in TSIMMIS. In *Proceedings of the AAAI Spring Symposium on Information Gathering, Stanford, California* (March 1995), pp. 61–64.
7. GOLGHER, P. B., LAENDER, A. H. F., DA SILVA, A. S., AND RIBEIRO-NETO, B. An Example-Based Environment for Wrapper Generation. In *Conceptual Modeling for E-Business and the Web, ER 2000 Workshops on Conceptual Modeling Approaches for E-Business and The World Wide Web and Conceptual Modeling*, S. Liddle, H. Mayr, and B. Thalheim, Eds., Springer, Berlin, Germany, 2000, pp. 94–101.
8. GUPTA, A., HARINARAYAN, V., AND RAJARAMAN, A. Virtual Database Technology. In *Proceedings of the Fourteenth International Conference on Data Engineering, February 23-27, 1998, Orlando, Florida* (1998), pp. 297–301.
9. LAENDER, A. H. F., RIBEIRO-NETO, B., DA SILVA, A. S., AND SILVA, E. S. Representing Web Data as Complex Objects. In *Electronic Commerce and Web Technologies, First International Conference EC-Web 2000*, K. Bauknecht, S. K. Mandria, and G. Pernul, Eds. Springer, Berlin, Germany, 2000, pp. 216–228.
10. MECCA, G., ATZENI, P., MASCI, A., MERIALDO, P., AND SINDONI, G. The ARANEUS Web-Base Management System. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (Seattle, Washington, June 1998), pp. 544–546.
11. RIBEIRO-NETO, B., LAENDER, A. H. F., AND DA SILVA, A. S. Extracting Semi-Structured Data Through Examples. In *Proceedings of the Eighth ACM International Conference on Information and Knowledge Management - CIKM'99* (Kansas City, Missouri, 1999), pp. 94–101.

Applied Information Security for m-Commerce and Digital Television Environments

Stefan Katzenbeisser and Philipp Tomsich

Institute of Software Technology, Vienna University of Technology
Favoritenstraße 9–11/188, A–1040 Wien, Austria
skatzenbeisser@acm.org, phil@ifs.tuwien.ac.at

Abstract. With the emergence of convergent information devices capable of delivering multimedia content and providing a network communication independent of location, new challenges regarding the creation of secure environments for conducting business electronically have arisen. Digital television (d-TV) uses high-bandwidth connections to provide on-demand pay-per-view multimedia content and value-added services, such as electronic shopping. Mobile electronic commerce (m-Commerce) extends these new business models into a wireless world. Both technologies share similar requirements in terms of information security, as they require proper authentication, message integrity and confidentiality of business contracts and personal preferences.

This paper presents protocols and infrastructure considerations to deal with the specific challenges arising from these applications: limited computational power, simple transfer of device ownership and transfer of access privileges between devices. The security requirements of future-generation electronic commerce applications are surveyed and protocols for use with these applications are discussed. In this context, trust-based authentication mechanisms (relying on recommendations and revocations) are adopted, in order to avoid static hierarchies and the need for central certification authorities.

1 Introduction

With the convergence of mobile communication networks and the Internet, e-commerce applications are increasingly used through mobile devices such as cell phones, PDAs (Personal Data Assistants) and palmtops. Mobile electronic commerce, in short *m-Commerce*, is utilizing these handheld, roaming devices for electronic commerce applications. Users can obtain information, browse product catalogs and place orders for (digital) goods, pay and contact customer support using handheld devices. Even real-time, multi-party price negotiations, such as in online auctions, have become feasible.

As m-Commerce is often coupled with novel business models, it also brings new challenges in providing information security, as information travels through multiple networks, often across wireless links and is processed by relatively powerless access terminals. However, most available solutions simply adapt security protocols designed for the Internet to mobile application scenarios. This poses several problems. One problematic issue in m-Commerce is authentication. Although applications (such as pre-paid mobile phones) exist which do not require proper authentication, more advanced applications will require authentication to set up legally binding constructs. In a “traditional”

approach, the user possesses a public/private key pair of any public key encryption algorithm, together with a certificate issued by a certification authority. If authentication is required, he engages in an authentication protocol relying on a public-key infrastructure. As long as any participant is working only on one device (and the secret key is stored there, protected by a passphrase), such an approach works well. However, the mobile world is more challenging, mainly because of two issues: First, to provide abuse of mobile devices by thieves, the user must authenticate himself towards the mobile device before starting any transaction. Second, in a mobile world where one user may have access to several ubiquitous mobile devices, one might end up with several public/private key pairs along with their certificates (e.g. in case one key pair is not transferable between devices due to software incompatibility). Setting up a traditional public key infrastructure with certification instances might be too expensive in such a scenario, keeping in mind the large number of devices which will eventually surpass the number of users by an order of magnitude, if the vision of ubiquitous network access using wearable computers becomes reality. Even if such a certification scheme can be implemented consistently, it will be difficult to keep the certificates up-to-date: e.g., imagine users selling their mobile devices to third parties or users replacing old devices with new ones.

Similar problems arise in electronic commerce applications for digital television (d-TV) environments. Tomorrow's television sets will contain multimedia terminals or be linked to set-top boxes which will not only provide Internet access, but also allow users to buy products advertised during a television show electronically. For example, during the telecast of a golf tournament, a small sign might pop up at the upper right corner of the TV screen, indicating that the viewer might buy some of the objects currently shown on the screen. Again it is hard to imagine that every digital television set in the world can be included in a traditional public-key infrastructure.

This paper discusses an infrastructure for e-Commerce and m-Commerce protocols, with an emphasis on keeping the protocols simple enough to use them on even the least powerful devices. Instead of using traditional public key certificates, the presented scheme relies on *trust tokens*, indicating that the issuer of the token believes to know the true identity of a customer (and a public key). In section 2, an e-Commerce scenario is introduced, which is used as an example in the remainder of the paper. Section 3 lists requirements for secure m-Commerce protocols and section 4 surveys related work which inspired the construction of "trust-based" m-Commerce protocols as introduced in section 5. Finally, conclusions are drawn and future research directions are outlined in section 6.

2 An Application: d-TV Commerce

A typical application could be similar to the following distributed application for delivering videos on demand in a digital television environment. It should be stressed that a similar scenario is applicable to m-Commerce applications.

Several content providers offer digital content like videos to customers, who are registered as users of a broadcast service offered by an infrastructure provider. Content providers use the infrastructure of the broadcast organizations to deliver their digital

content to the customers. Customers can order products either directly from the content provider or by using the infrastructure of the broadcaster.

Several questions concerning computer security can be raised immediately. As a purchasing order of a customer has to be legally binding, authentication between the customer and the content provider is needed. This is also a necessary prerequisite for fair accounting schemes (i.e. only the actual consumer of a product or service is charged). Furthermore, control messages sent between all parties must be confidential and no party should be able to alter it during transmission without means of detection. Both the customer and the content provider should not be able to falsely deny later their agreement on the purchasing conditions. Another requirement may be anonymity, as customers will not fully accept services that erode their privacy. For example, in a digital television environment, analyzing viewing habits is simplified and customer profiles can be compiled for direct marketing campaigns.

Electronic commerce protocols also need to include some sort of copyright protection mechanism in case digital objects are sold electronically, to guarantee acceptance by content creators and thus ensure a wider range of quality content [8].

3 Application Requirements

In contrast to other electronic commerce applications (employing general purpose personal computers and web browsers), partially hardware-based security solutions are often feasible in m-Commerce applications. It may be assumed that every user is equipped with at least one mobile device that contains a unique identification string called Hardware-ID (e.g., a string containing the name of the manufacturer and a serial number or MAC address). Furthermore, every user possesses a smartcard containing a unique number (a Customer-ID) and an expiration date in non-volatile, tamper-proof memory; this card will be used to authenticate the user towards the mobile device. It is assumed that the broadcaster issues such identification cards to its customers and updates them in case they have expired; these cards are not transferable between users. The mobile device will not allow any transaction before the user engaged in an authentication protocol.

In such an environment, m-Commerce protocols must be constructed that rely on a decentralized authentication mechanism. One of the major design criteria is to minimize the cost of maintaining a certification instance, as it is likely that m-Commerce users change their mobile devices often. Ideally, the infrastructure should provide a mechanism that keeps the amount of customer interaction as minimal as possible in case a device is registered or unregistered. Such a protocol is described in section 5.

The following security requirements for e-Commerce applications [7] can be identified:

- **Confidentiality.** Persons not involved in an electronic transaction should not be able to gain any information about the orders placed by a user at a content provider; thus, all messages sent in the protocol must be encrypted.
- **Authentication.** As contracts must be legally binding, an authentication mechanism for both customers and content providers is required.
- **User Authentication.** To prevent theft of mobile devices, the user has to authenticate himself to the device before starting any transaction. However, this authentication

overhead must be as low as possible; otherwise the usability of the mobile device might be affected.

- **Integrity.** Messages sent in the electronic commerce protocol must not be altered by any party not involved in the transaction.
- **Fairness.** m-Commerce protocols may include some kind of copyright protection mechanism. Such a mechanism must be “fair” towards both the customer and content provider, meaning that no false claims of infringements should occur. However, in case an illegal copy is found, it must allow resolving the copyright situation.
- **Non-repudiation.** No party should be able to deny later their agreement to the conditions of any given transaction.
- **Partial anonymity.** In order to provide privacy of the customers, anonymous transactions are necessary. However, since this is nearly impossible in reality, partial anonymity will be required, which distributes the relevant information in such a way, that the creation of a complete customer profile by any participating party without cooperation becomes unlikely.
- **Partial Traceability.** In case one user performs illegal actions, it must be possible to retrieve that person's identity. This is complicated by the partial anonymity requirement, but possible in a fashion similar to the tracing of email: every party knows “someone” who is closer to knowing the real identity of a given party; by following these links, a party with authoritative information on a user's identity will be eventually found.

Clearly, these requirements could be met by using public-key cryptography relying on a certification mechanism and copyright protection techniques like watermarking [4]. However, the possibility of using only “light-weight” security features has been explored in the protocols presented here. Although the level of security can not match that provided by traditional mechanisms, it reduces the overhead costs of maintaining a public key infrastructure. In general, a *sufficient level of security for conducting business at minimal cost for maintaining the infrastructure* is the motivating factor.

The design of m-Commerce protocols is strongly influenced by the available hardware and its computing power (e.g., the hardware in set-top-boxes might contain specialized processors for processing multimedia streams, but miss the required general purpose hardware for the implementation of a public key cryptosystem), the existence or absence of a traditional public-key infrastructure and the organizational constraints of the content providers and broadcasters (e.g., requirements for accounting between the content provider and the broadcaster). Central to any successful protocol is, that it provides security features in a transparent way, so that a minimum amount of user interaction is required.

4 Related Work

Several related publications can be identified that discuss parts of the problems mentioned above. The problem of user authentication was addressed by Ebringer et al. [2] by a technique called *parasitic authentication*, which uses a second mobile device (like a smartcard) for authentication purposes. As long as the mobile device is able to communicate with the secondary device and verify that it indeed is the same device that

was used in a setup process, the mobile device allows transactions. Depending on the computational power of the secondary device, several authentication mechanisms could be implemented. In its simplest form, the secondary device is strictly passive, storing only some identification string; this form is, however, not more secure than a simple password mechanism. In case the device is able to retrieve and store several passwords, a computationally secure system can be implemented. If the secondary device is even able to compute hash functions or perform modular arithmetic, traditional authentication schemes such as the Schnorr identification scheme can be adopted. However, the cost of such secondary devices grows rapidly with the computing power they provide.

In case the mobile device is not able to provide the computation power needed in electronic transactions (e.g. if the device does not contain a programmable general-purpose processor, but only specialized hardware), it is possible to use server-aided signature generation protocols. Instead of adding expensive hardware-based solutions for generating digital signatures, the mobile device lets a more powerful server perform the computationally expensive parts of signature generation, without giving away enough information to reconstruct the secret key [5].

The work on a formal trust model [1] greatly inspired the construction of trust-based protocols, as introduced in the next section. Trust is defined as the particular level of the subjective probability with which an agent will perform a specific action; trust is propagated through a distributed system by direct or indirect recommendations. For example, agent *A* might have direct information on agent *B* and might be able to assess his trustworthiness. In case no direct information is available, agent *A* can query agent *C* who might himself be able to get information on *B*; he provides this information to *A* by using a *recommendation*. In case trust is assumed to be transitive, *A* can assess the trustworthiness of *B* with the help of *C*; however, the trust level might be lower, as no direct information is available. In case more recommendations are available, *A* is able to compute a trust level for each of them and base his final decision on the average of all trust levels of all available recommendations.

A similar approach was taken by Rasmusson and Jansson [6], who proposed “soft” security mechanisms for electronic commerce; a marketplace consists of two types of agents: buyers and sellers. It is up to the agents to collect information about “cheaters” and propagate this information through the system. We will use a similar construction for authentication purposes in a mobile electronic marketplace.

5 Trust-Based d-TV and m-Commerce Protocols

In this section we introduce *trust-based authentication* methods. Suppose that a distributed system consists of n agents, divided in customers, content providers and broadcasters or infrastructure providers. In order to make legally binding transactions possible (this includes tracing of illegally behaving users) while retaining the privacy of customers, the mapping between one Customer-ID and Hardware-ID, together with a public key of the customer, must be known by content providers. In case one agent in the system believes to know the binding between a Customer- and Hardware-ID, he issues a *trust token* (which is also called *recommendation*). Syntactically, such tokens contain a Hardware-ID, a Customer-ID and a public key of any public-key cryptosystem, together

with an expiration date and trust level, all signed by one agent. A token indicates that the signing agent believes to know both the public key of a customer and the mapping between his Hardware- and Customer-ID up to a certain extent, indicated by the trust level. Tokens expire after a specific amount of time; in the simplest form, trust levels are real numbers between 0 (complete distrust) and 1 (complete confidence in the identity of a given user). The necessity for expiring entries results from the high number and frequency of recommendations that may be issued and may pollute the trust database as well as make the determination of an overall level of trust difficult.

In order to provide partial anonymity and to make recording of user habits more difficult, special agents, called *trust publishers*, are used to record trust tokens in a distributed manner, thereby making them publicly available. In case a content provider wants to start a transaction with a customer, he queries all known trust publishers for recommendations; based on the trust levels of the received trust tokens and the trustworthiness of the trust publishers, he bases his decision whether or not to start the transaction (i.e. he believes to know the true identity of the customer). While a traditional public key infrastructure is used to authenticate trust publishers and the content providers, authentication between mobile devices is entirely based on a distributed architecture that does not involve central trusted parties. As most transactions are done locally (i.e. within one country or even within one city), we believe that the overhead of querying several trust publishers is not significant. In case one party detects that one recommendation falsely identifies one Hardware-ID and a Customer-ID (or in case a token contains an incorrect public key), he sends a *revocation*.

5.1 Recommendation and Revocation Protocols

In case one agent believes to know the mapping between one Customer-ID and Hardware-ID and the customer's public key (e.g. after the successful completion of a commerce transaction or a registration process), he engages in a recommendation protocol. A new trust token is constructed which contains both ID's and the customer's public key. The agent then proceeds to assess the trust-worthiness of the ID-mapping. In case personal contact with the customer existed and an identification card was shown, the confidence level will be high—if the transaction is done electronically or via phone, the confidence will be lower. The agent signs the token and forwards it to one or more trust publishers. Note that the user name is *not* published in the token. The trust publisher only receives the mapping between a Customer-ID and a Hardware-ID. Without help of the agent he is not able to uniquely identify a specific person knowing the Customer-ID (or a public key) only.

The trust publishers verify the signature on the trust token. In case it is valid, the signature is stripped off and the token is re-signed by the publisher. This provides *partial anonymity*, as it is not possible by a third person to identify the issuer of the token without help of the trust publisher. Such a process significantly complicates the compilation of user habits by involving multiple parties. Tokens are published, together with an assessment of the reliability level of the source of the token (based on the trust publisher's recent experiences). Both the reliability and trust levels can be used in assessing the quality of the trust token in future transactions. *Partial traceability* is ensured by recording the issuer's identity for each trust token within the records of the trust publisher (this information is not public).

In case any agent is detecting a breach of trust, a *revocation* token is sent to all known trust publishers. The revocation contains again both IDs and is signed by the party who detected the breach of trust. Trust publishers check the validity of the signature, delete all recommendations containing the same IDs contained in his database and publish the revocation in a partially anonymized manner.

5.2 Authentication Request

If an agent wants to verify a mapping between a Hardware-ID and a Customer-ID or the validity of a public key (in case a transaction with a specific customer is to be initiated), a query to several trust publishers containing both IDs needs to be issued. In case a revocation token is received, indicating that a previous recommendation was canceled, the transaction is aborted, as the mapping between the IDs is likely to be false (in this case the agent may require that another agent checks the true identity of the customer and issues a new trust token). Otherwise, the agent collects all trust tokens and computes a derived trust value from the trust values contained in these tokens and the associated reliability estimates (according to section 5.3). The decision on whether or not a transaction is started is based on the trust value resulting from this computation (i.e., only if the value is larger than some minimal trust threshold, a transaction is actually started). In case an assessment of the reliability of the trust publishers is available, it can also be incorporated in the final trust value calculation.

After a successful transaction, an agent may again issue a recommendation, advising that a certain combination between a Customer-ID and Hardware-ID seems to be correct. However, we should note that the only purpose of a trust token is to establish a binding between a Customer- and a Hardware-ID and to transfer a key; it should *not* be used for transmitting trust concerning the owner himself, such as information about a client's credit history. If no valid recommendations can be found, an agent may require the customer to re-authenticate with a different agent (e.g., the local infrastructure provider). In case the business model of an agent allows taking some risks, he can continue the transaction without proper authentication and rely on the legal system in case of abuse (the provider may then issue a new trust token with low trust level when finishing the transactions). In order to avoid this case entirely, one could also adopt a technical solution: one specific token issued by a local infrastructure provider does not expire, but will be marked invalid in case a revocation is sent by an agent of the system.

The proposed authentication mechanism makes tracing of user habits difficult; it would require collaboration of at least three parties: a content provider, the trust publisher and a third party who wants to collect information about a customer. In order to retrieve personal information, the following steps are necessary: Given one trust token the information harvester collaborates with the trust publisher publishing this specific token, to retrieve the identity of the issuer of that token. In the next phase, cooperation with the token's issuer is necessary to reveal the true identity of the customer. Due to the distributed nature of the protocol, collaboration with a large number of partners becomes necessary to retrieve information on a customer.

Unfortunately, the system is susceptible by a denial-of-service attack in case one agent issues several trust tokens with a very low trust level or falsely sends a revocation. In this case, the final trust value computed by another agent could be so low that he decides not to start a transaction with a specific customer. Such attacks have to be prevented by

organizational means; in case such an attack is detected by one trust publisher, the attacker is marked as unreliable in the records of the trust publisher, thereby decreasing the impact of its recommendations in the final trust value computations and effectively excluding an agent from any further *active participation* in the recommendation and revocation process.

5.3 Trust Value Semantics

One primary benefit of the system is that all agents can behave autonomously. Instead of relying on a central trusted authority to validate consumer identities, the agents can act according their own decision rules and believes. Specifically, agents are faced to solve the following problem: given n trust tokens containing trust levels t_i , which were rated by the trust publishers with reliability levels l_i , the agent has to compute a final trust value t . This value is to be put in relation to a minimal trust threshold t_0 required by an agent's company policy.

In a simple model (more sophisticated metrics [3] are available for various purposes), both l_i and t_i are real numbers in the interval $[0, 1]$. Whereas 0 describes complete distrust or unreliability, 1 stands for the highest trust or reliability level. One decision rule could define the final trust level t as a weighted mean of the t_i ,

$$t = \frac{\sum_{i=1}^n l_i t_i}{\sum_{i=1}^n l_i}.$$

In this equation, l_i determines the impact of the i -th trust token on the final verdict. Further, an appropriate bound $t_0 \in (0, 1)$ on t has to be chosen. Intuitively, t_0 models an agents ability to accept business risks. However, there is no need that every agent uses the same decision procedure. If information on the reliability of the trust publisher is also available, an agent might introduce a reliability measure for any information received from that trust publisher and factor it into the calculation.

5.4 Trust-Based Authentication in d-TV or m-Commerce Scenarios

Doing business in an electronically mediated environment, such as m-Commerce, introduces additional uncertainty in regard to customers' identities. Consequently, before permitting the first transaction, a registration of the new customer with the broadcaster or the infrastructure provider becomes necessary to fully establish a user's identity once. In order to communicate the user's identity in a partially anonymized way to partners, some credentials are issued to new customers (a smart card could be issued containing a Customer-ID). An initial trust token for the customer will be generated at this point and broadcast to trust publishers and also stored in the permanent memory of the user's access terminal. Additionally, the unregistration of customers and a process to transfer ownership of an access terminal are necessary:

- **Registration.** In order to issue the initial trust token and guarantee at least a partial traceability, potential customers need to identify themselves to an infrastructure provider or a similar market participant. Traditionally, this is done by entering a store and providing one's identification card. Alternatively, registration could be

done electronically (particularly by existing customers; e.g. if an additional access terminal is acquired) using passwords, transaction codes or similar mechanisms. During the registration, the hardware device generates a public/private key pair; the public key will be included in the trust token issued by the infrastructure provider. Furthermore, every device will contain an *access control list*, i.e. a list of users (Customer-ID's) that will be allowed to use the device for commerce transactions. The infrastructure provider stores the Customer-ID within the access control list of the terminal to reflect ownership and hinder theft.

- **Electronic Transaction.** An electronic transaction involves a customer and a content provider and is carried using the network infrastructure of the broadcaster. It consists of four distinct steps: user authentication, trust-based customer authentication, e-commerce transaction protocols and updates to the trust publishers.

User authentication is designed to prevent the theft of mobile devices and other access terminals. The customer uses his smartcard to authenticate himself to a given device. The device will only work for users that match its access control list. Depending on the computing power of the smart card, this can either be a simple comparison of an ID stored on the smartcard and in the device. Alternatively, one could adopt parasitic authentication [2]. If an unauthorized smartcard is used, the device locks itself and forces the user to re-register (i.e., to simply use a smartcard that is on the access control list will not unlock the device again).

Trust based customer authentication uses the protocol introduced in the previous sections. It is used by a content provider to establish a confidence in whether a customer is who he pretends to be. For this purpose, the customer sends both his Customer-ID and Hardware-ID, along with the public key of the device, which are used in a query to trust publishers. This serves to establish a numeric representation of how likely the Customer-ID and Hardware-ID match, without requiring a public key infrastructure for the customers. The content provider can then use the public key of the customer (contained in the trust tokens) in future transactions.

The *execution of e-commerce protocols* takes care of selecting and exchanging the goods. This will include negotiation price-building and copyright protection mechanisms, such as watermarking and fingerprinting of digital content to trace illegal copies [48].

After a successful transaction completion or a breach of trust (i.e. the user's identity is established to be incorrect), *updates to the trust publishers* will be sent out to reflect the newly gained information. This is done either in form of a recommendation or in the form of a revocation.

- **Unregistration.** If a customer decides to discontinue a subscription from the local infrastructure provider, an unregistration is necessary. During this process, a revocation of trust is sent out to known trust publishers (trust publishers are likely to propagate revocations to other trust publishers). The trust token is deleted from the access control list of the customer's access terminal. A special case of unregistration is the sale of the access terminal by the customer: when the former owner unregisters his device, the device is reset into an "unowned" state and needs to be bonded to the new owner using a (simplified) registration process.

6 Conclusions and Future Research

The security requirements for secure electronic commerce transactions in a mobile world, which is characterized by a large number of ubiquitous devices, have been discussed. Based on an overview of requirements and applicable protocols, a trust-based authentication model has been introduced to eliminate the need for a large-scale public key infrastructure, especially for m-Commerce and d-TV applications.

Future research is required on some of the practical aspects of implementing such a trust-based infrastructure: *alternative formulas* for the calculation of the derived trust value are an apparent area of improvement; *adding a feedback mechanism* that allows to modify trust values of trust tokens after they have been issued; *a peer-to-peer synchronization* of trust-publishers may help to pass initial trust tokens issues by infrastructure providers more quickly.

Generally, we believe that a distributed architecture is superior to systems involving central trusted certification authorities both in terms of the effort needed to maintain such an infrastructure and in the increased privacy of individual users.

References

1. A. Abdul-Rahman, S. Hailes, "A Distributed Trust Model", in *1997 New Security Paradigms Workshop, Proceedings*, ACM Press, pp. 48–60, 1998.
2. T. Ebringer, P. Thorne, Y. Zheng, "Parasitic Authentication To Protect Your E-Wallet", in *IEEE Computer*, vol. 33, no. 10, pp. 54–60, 2000.
3. A. Jøsang, "A Subjective Metric of Authentication", in *Proceeding of the 5th European Symposium on Research in Computer Security*, Springer Lecture Notes in Computer Science vol. 1485, pp. 329–344, 1998.
4. S. Katzenbeisser, F.A.P. Petitcolas (eds.), *Information Hiding Techniques for Steganography and Digital Watermarking*, Boston, London: Artech House, 2000.
5. T. Matsumoto, K. Kato, H. Imai, "Speeding up computation with insecure auxiliary devices", in *Advances in Cryptology, Proceedings of Crypto '88*, Springer Lecture Notes in Computer Science vol. 403, pp. 497–506, 1989.
6. L. Rasmusson, S. Jansson, "Simulated Social Control for Secure Internet Commerce", in *1996 New Security Paradigms Workshop, Proceedings*, ACM Press, pp. 18–25, 1997.
7. D. V. Thanh, "Security Issues in Mobile eCommerce", in *Electronic Commerce and Web Technologies, First International Conference, Proceedings*, Springer Lecture Notes in Computer Science vol. 1875, pp. 467–476, 2000.
8. P. Tomsich, S. Katzenbeisser, "Towards a Secure and De-centralized Digital Watermarking Infrastructure for the Protection of Intellectual Property", in *Electronic Commerce and Web Technologies, First International Conference, Proceedings*, Springer Lecture Notes in Computer Science vol. 1875, pp. 38–47, 2000.

Flexible Authentication with Multiple Domains of Electronic Commerce

Kyung-Ah Chang, Byung-Rae Lee, and Tai-Yun Kim

Dept. of Computer Science & Engineering, Korea University,
1, 5-ga, Anam-dong, Sungbuk-ku, Seoul, 136-701, Korea
{gypsy93, brlee, tykim}@netlab.korea.ac.kr

Abstract. In this paper, based on CORBA security service specification[1, 3], we propose the authentication model supporting multiple domains for electronic commerce with an extension to the Kerberos[13] authentication framework using public key cryptosystem[15]. This proposed model supports the protection of the high-level resources and the preservation of the security policies of the underlying resources that form the foundation of various domains, between the Kerberized domains[14] and the Non-Kerberized domains. Also we achieved the flexibility of key management and reliable session key generation between the Client and the Provider using the public key cryptosystem.

1 Introduction

The technological advances in recent years have led to a situation where large, distributed applications that cooperate with each other are becoming an essential part of IT technology for Electronic Commerce(EC). The emerging widespread introduction and the growing concern as the Internet have the potential to transform the way people compute, and more importantly the way they interact and collaborate with one another. However, in the face of the onrush of hardware, the community has tried to stretch an existing paradigm into a regime for which it was not designed.

The traditional requirements of security mechanisms and policies are exacerbated in the current distributed computing, as the physical resources of this exist in multiple administrative domains, each with different local security requirements. Much attention has been devoted to security issues and it is apparent that a high level of security is a fundamental prerequisite for Internet-based transactions, especially in the EC area.

As a consequence, the need for standard architectures and frameworks for developing such applications has arisen. The OMG [1] has specified the CORBA in response to these needs[2, 4, 6]. CORBA[8, 9] is a standard middleware supporting heterogeneous networks, designed as a platform-neutral infrastructure for inter-object communication. The benefits of CORBA are that the entire system is self-describing, that the interoperability[5] is able to support construction of larger integrated components using existing components, and that the specification of a service is always separated from the implementation.

However, CORBA based security service specification[1, 10] itself does not provide any security mechanism[6]. The predefined attributes of CORBA security serv-

ice have only limited validity and many security mechanisms do not provide sufficient security attributes.

This paper, based on CORBA security service specification[1], proposes the authentication service model supporting multiple domains with an extension to the Kerberos[13] authentication framework using public key cryptosystem(PKC)[15]. This proposed model, by public-key certificates, assures the identification of a partner in the authentication of peer entities and the secure access to multiple domains in the authorization of underlying resources. Since our deployed Kerberos is extended to the authentication service model, it provides the flexibility of key management and the ability to leverage the public key certification infrastructure[7].

The organization of this paper is as follows. Section 2 presents the description of authentication provider in this paper and the CORBA security service. Section 3 describes the structure of the flexible authentication service for multiple domains in detail. Finally, Sections 4 and 5 contain a performance and conclusion, respectively.

2 Security in Object Based EC System

2.1 Authentication Provider Supporting EC

On Internet, Certificate Authorities(CA) acts as trusted intermediaries when authenticating clients and servers. Authentication Provider in our proposed service is another form of trusted intermediary[14]. In a public key scheme, a CA issues a long-lived credential - a public key certificate. When both clients and servers have such certificates they can authenticate to each other without further reference to a CA. However, precisely because these certificates are long-lived, some method is required to inform servers of revoked certificates. This can be done by requiring servers to check a certificate's current validity with the CA on each use of a certificate, or by distributing Certificate Revocation Lists to all servers periodically. In the proposed scheme, the Authentication Provider issues clients a short-lived credential, which must then be presented to obtain an access right for a particular server.

The Authentication Provider described in this paper is structured in layers[16, 17]; Exchange Layer, Supporting Services Layer. The Exchange Layer and Supporting Services Layer are responsible for the execution of CORBA security service by adding of a security and a message interceptor[3]. The Supporting Services Layer provides persistent object storage, a communication, and a cryptographic service.

The Exchange Layer provides services for handling and packaging business items as well as transfer and fairness of mutual exchanges. The security attributes stored in each type of the basic objects determine the label of privilege that is required for the exchange.

We concentrate on the Policy Enforcement Block(PEB) that handles a credential of each participant, thereby performing all secure invocations between a Client and a Provider. This block contains Authentication Service, Authorization Service, and Session Identifier Service.

2.2 CORBA Security Service

The CORBA security service specification[1] is large in part due to the inherent complexity of security, and due to the fact that the security service specification includes security models and interfaces for application development, security administration, and the implementation of the security services themselves. All these and their interfaces are specified in an implementation of independent manner[9, 10]. So the interface of security service is independent of the use of symmetric or asymmetric keys, and the interface of a principal's credential is independent of the use of a particular certificate protocol. Fig. 1 shows a brief mechanism of CORBA based security service.

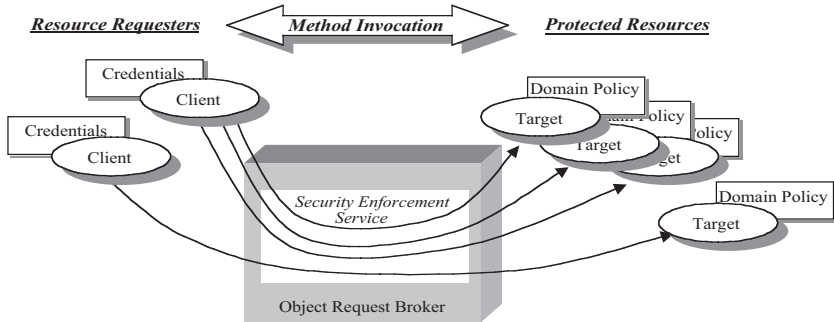


Fig. 1. CORBA based Security Service

The objective of this specification[1, 2] is to provide security in the ORB environment in the form of an object service. The focus lies hereby on confidentiality, integrity, and accountability. The model used by the CORBA security service specification involves principals that are authenticated using a principal authenticator object. Once authenticated, a principal is associated with a credential object, which contains information about its authenticated identity and the access rights under this identity. These credentials are then used in secure transactions, to verify the access privileges of the parties involved, and to register identities for actions that are audited or carried out in a non-repudiation mode.

3 Flexible Authentication Service for Multiple Domains

Current distributed computing based EC is really a federation of resources from multiple administrative domains, between the Kerberized domains[14] and the Non-Kerberized domains, each with its own separately evaluated and enforced security policies. The security service of these environments must protect the high-level resources and preserve the security policies of the underlying resources that form the foundation of various domains.

In this paper, the assumption of our scheme is that only objects of users that have been authenticated must be authorized to use the underlying resources over multiple domains. And assume that system administrators allow their systems to participate in our authentication service.

With authentication() method, the US initiates the authentication exchange by requesting the Session Identifier(SID_{as}) from the Authentication Service Block(AS). This is necessary since the construction of the subsequent Authorization Service Phase requires the certificate of Provider($CertP$).

At the start of the protocol, there is one of the assumptions that AS has long-term secret and public key agreement keys v and g^v . Another assumption is that US possesses the public key necessary to verify certificates issued by Authentication Service Provider. In first request message, as known in [18, 19], US generates a random number u and computes temporary public key agreement key g^u . The US then generates an encryption session key $L = (g^v)^u$ where g^v is the public key agreement key of the AS.

On receipt of the first message, as shown in (a) of Fig. 2, AS does not know with whom he is communicating. AS computes $L = (g^u)^v$ and generates a session key $K_{u, azs}$ between US and Authorization Service Block(AZS). He then sends to US message encrypted using L together with Session Identifier(SID_{azs}) to access AZS encrypted using the secret key of the AZS, K_{azs} .

Once the US has obtained SID_{azs} , it implies being authenticated by an authentication object, and can proceed to generate SID_p for service request. The message contains similar information to that in a traditional ticket request of Kerberos.

The method authorization() handles a Session Identifier(SID_{azs}), the Provider(id_p) that the Client wants to access, and the name of the operation to invoke. We get Provider's name(id_p) from the object storage of Kerberized hosts and the principal(id_u) from the Session Identifier(SID_{azs}). If the name of the Provider(id_p) the Client wants to invoke is among these, this is allowed to proceed. If the authorization succeeds, the operation and the returns are subsequently invoked on the Provider. If not, the CORBA system exception[8] 'NoPermission' is flagged.

At the start of this phase, as shown in (b) of Fig. 2, there is an assumption that AZS has kept key escrow of Provider and has shared the domain(Domain_p) with Provider. In first request message, US send to AZS Provider's identity(id_p) with SID_{azs} encrypted using secret key of AZS(K_{azs}).

On receipt of the first message in this phase, AZS decrypts the message using his secret key. It then retrieves the session key between US and AZS, which is found in the SID_{azs} . And then he generate the appropriate certificates required in the protocol($CertU$, $CertP$) and the Session Identifier(SID_p) to access Provider. He send to US theses messages together with Session Identifier(SID_p) encrypted using the secret key of the Provider(K_p).

The Session Identifier(SID_p) received by the AZS is simply a conventional service ticket. At the start of this phase, as shown in (c) of Fig. 2, US has Provider's public key g^p in the certificate $CertP$ and Client's private key w in the certificate $CertU$. And then he computes an encryption session key $K_{u, p} = (g^p)^w$. In first request message, US send to PS the certificate of client($CertU$) and SID_p and the Authenticator which additional data needed as input to the payment scheme with encrypted using encryption session key $K_{u, p}$.

On receipt of the first message, PS computes an encryption session key $K_{u, p} = (g^w)^p$ then decrypts the encrypted Authenticator message. All operations from this point on can protocol per normal Kerberos operations.

3.2 Multiple Access Supporting Kerberized Domains

On Kerberized domain, our client's request of Authentication Service Model needs initial objects to reference the security service, like it does for other CORBA based services. These objects are 'SecurityLevel2::UserSponsor(or ProviderSponsor)' and 'SecurityLevel2::Current'[8, 10]. The references to these locally constrained objects

The diagram illustrates the ORB Core architecture, showing the interaction between a Client and a Provider through a central ORB Core.

Client Side (Left):

- Client application access decisions:** Includes Exchange Mng. and Credentials.
- User Sponsor:** Interacts with the Client application access decisions and the "Current" object.
- Client Policies:** Interacts with the "Current" object.
- "Current" object:** Acts as a central point for the Client's interaction with the ORB Core.
- Client Authenticated ID:** Includes Identity attribute and Privilege attribute.
- PEB (Policy Enforcement Block):** Contains AS (Access Service) and AZS (Access Zoning Service).
- Access Control:** Interacts with the AS and AZS.
- Client access decision:** Includes Access Policy and Required Rights.
- ORB Security Services:** Interacts with the AS and AZS.

Provider Side (Right):

- Provider application access decisions:** Includes Exchange Mng. and Credentials.
- Provider Sponsor:** Interacts with the Provider application access decisions and the "Current" object.
- Provider Policies:** Interacts with the "Current" object.
- "Current" object:** Acts as a central point for the Provider's interaction with the ORB Core.
- Provider Authenticated ID:** Includes Identity attribute and Privilege attribute.
- PEB (Policy Enforcement Block):** Contains AS (Access Service) and AZS (Access Zoning Service).
- Access Control:** Interacts with the AS and AZS.
- Provider access decision:** Includes Access Policy and Required Rights.
- ORB Security Services:** Interacts with the AS and AZS.

ORB Core:

- The ORB Core is the central component that facilitates the interaction between the Client and the Provider.
- It includes the ORB Security Services and the Access Control component.
- The ORB Core is responsible for enforcing the access decisions and policies.

The Credential has to be created from its own certificate. This object holds the security attributes of a principal, e.g. authenticated identity and privileges. It is used by the application to create its own security information that later should be sent to the remote peer during the establishment of the secure association between the Client and the Provider.

UserSponsor(or ProviderSponsor) is responsible for managing the placement, activation, and deactivation of set of objects. They provide a central mechanism for specifying policy for a set of like objects. For setting policy for instances, UserSponsor serves as location authorities for instances, supporting the binding of secure association.

Finally, as shown in figure of previous section, it can establish a fairness of exchanges through the Authentication Service Provider using items bundled as a

CORBA object. Both the UserSponsor and the ProviderSponsor transfer Session Identifiers(SID_p) before mutual exchanges occur. This prevents the Current from being transferred Current to illegal peers, and prevents the Clients from being given access to illegal peers.

3.3 Flexible Multiple Access Supporting Multiple Domains

For multiple domains, all communication of our model is done via Kerberos mechanisms. Thus cross-realm authentication[13] is immediately and transparently supported: UserSponsor only has to be performed once for each group of Kerberos realms that support cross-realm authentication with each other. The initial objects will automatically obtain SID_{azs} s for the other realms based on the existence of a valid SID_{azs} for a given host.

Our model is assumed with Kerberized domains basically, however, we must consider multiple domains, specifically integration of Kerberized(K_{sys}) and the Non-Kerberized domains(NK_{sys}). We propose the approach to issue temporary credentials to NK_{sys} 's request objects[11, 12]. Most of all participated NK_{sys} involve a single CA.

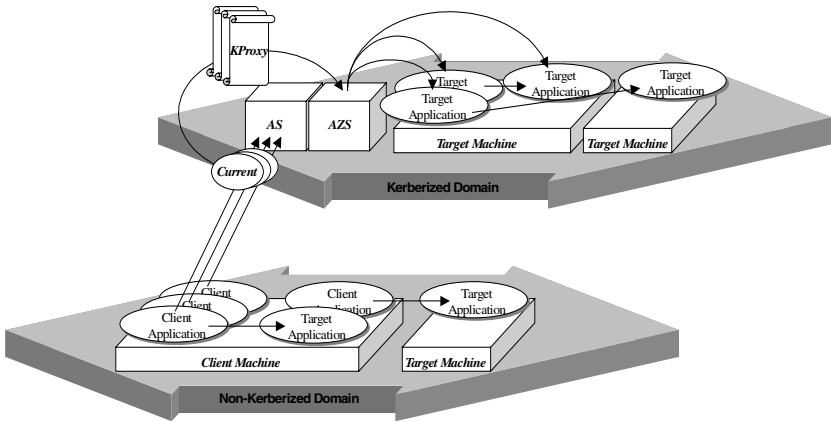


Fig. 4. Authentication mechanism of multiple domains access

In general, users in one domain are unable to verify the authenticity of certificates originating in a separate domain. However, our K_{sys} 's Authentication Service Provider issues a cross-certificate[4] based temporary certificate. A trust relationship between CAs of multiple domains and our Authentication Service Provider must be specified in order for users under CAs to interoperate cryptographically.

As shown in Fig. 4, the essential component of our approach is a $KProxy$ object for each NK_{sys} client's request. This $KProxy$ securely holds the delegated credentials as K_{sys} client's in the local Kerberized system. Whenever the client's request from the NK_{sys} wants to create an object on client's behalf on its associated physical machine, the Authentication Service Provider creates the Current object that contains minimal permission. Provider's AS will only issue client's delegated credentials of that domain if client's valid temporary certificates are presented in the request. A delegated credential specifies exactly who is granted the listed rights, whereas simple posses-

sion of a bearer credential grants the rights listed within it. Then the Current performs a call back to the *KProxy* for client to obtain a SID_{azs} for that particular Provider.

After obtaining a SID_{azs} , Provider's AZS is obtained the attributes from the *KProxy* object by calling 'get_attributes'. Then all operations from this point on can mechanism per Kerberized operations for multiple domains.

The access control mechanisms of client's *KProxy* can be configured to issue the thread specific Credential obtaining the information to the thread of request execution. There might be more than one thread, every thread associated with a different set of security attributes. These can be accessed from the appropriate Current has to be used.

4 Security Analysis

4.1 Analysis with our Security Scheme

In distributed computing, the DCE[20] and the SESAME[21, 22] are the well-known security systems, based on Kerberos, of the Client/ Server architecture. The OSF/ DCE provides security for distributed applications and it also provides secure access into and between the DCE services. In the basic the DCE environment, an application server that is the reference monitor for the resource manages access to resource. The SESAME technology offers sophisticated the Single Sign-On with added distributed access control features and cryptographic protection of interchanged data.

As seen in the Table 1, our proposed security scheme better security features like the authentication for peer entities to perform the mutual negotiations and the fairness of exchange. Therefore it has the good advantages of interoperability with other security services.

Table 1. Analysis of Our Security Scheme

	DCE	SESAME	Our Security Scheme
Access control level	Application	Application	Application/ System
Authentication	Unilateral	Unilateral	Unilateral/ Mutual
Authorization policy	ACL based	ACL based	Label based Rule
Fairness of exchange	No	No	Yes
Flag of privilege type	Positive	Positive/ Negative	Positive
Grant/ Revoke privileges	Controlled by Server	Controlled by Server	Label based Rule
Scalability	Average	Average	High
Security policy domain	Server's domain	Server's domain	System Imposed
Suitability	Stable User base	Stable User base	Mandatory Controls

And then Table 2 shows comparison of key distribution scheme with traditional Kerberos[13]. Our proposed security scheme provides session key establishment mechanism based on PKC.

Table 2. Comparison of Key Distribution Scheme with traditional Kerberos

	Kerberos	Authentication Service Model
Session Key Establishment between User and TGS(AS)	Symmetric Key Transport	ElGamal
Session Key Establishment between User and SGS(AZS)	Symmetric Key Transport	Symmetric Key Transport
Session Key Establishment between User and Service Provider	Symmetric Key Transport	ElGamal
Role of TGS(AS)	Symmetric Key Transport for User and Provider	Public Key Certificate Generation

5 Conclusion and Future Works

We have proposed the CORBA based authentication service, and within that service we have presented flexible mechanisms to accommodate multiple domains. The goal of our system is to select resources for use by applications and securely coordinate execution of application in multiple domains, eliminate the need for the end-user of EC to explicitly log on to each machine.

The authentication service that proposed in this paper is very critical in EC, since it supports the protection of the high-level resources and the preservation of the security policies of the underlying resources that form the foundation of various domains, between the Kerberized domains and the Non-Kerberized domains. Using public-key cryptosystem we acquired the flexibility of key management and reliable session key generation between the Client and the Provider.

Research should be made on the efficient distributed object system to support a distributed security for EC, and offer a more elaborated security infrastructure. In addition, for a key management, a heterogeneous key distribution of session keys should be considered.

References

1. OMG, CORBA services: Common Object Security Specification v1.7 (Draft), <ftp://ftp.omg.org/pub/docs/security/99-12-02.pdf>, 2000.
2. Object Management Group. CORBA/ IIOP 2.3.1 specification, <http://sisyphus.omg.org/technology/documents/corba2formal.htm>, 1999.
3. OMG Security Working Group, OMG White Paper on Security, OMG Document, No. 9, 1996.
4. Menezes, Van Oorschot, Vanstone, Handbook of Applied Cryptography, 2nd Ed., pp.570-577, 2000.
5. OMG, Common Secure Interoperability V2 RFP, http://www.omg.org/techprocess/meetings/schedule/Common_Secure_Interop_V2_RFP.html, 2000.
6. A. Alireza, U. Lang, M. Padelis, R. Schreiner, and M. Schumacher, "The Challenges of CORBA Security", *Workshop of Sicherheit in Mediendaten*, Springer, 2000.
7. DSTC, Public Key Infrastructure RFP, <ftp://ftp.omg.org/pub/docs/ec/99-12-03.pdf>, 2000.

8. Robert Orfali, Dan Harkey, *Client/ Server Programming with JAVA and CORBA*, John Wiley & Sons, 1997.
9. Andreas Vogel, Keith Duddy, *Java Programming with CORBA*, 2nd Ed., John Wiley & Sons, 1998.
10. Bob Blakley, *CORBA Security: An Introduction to Safe Computing with Objects*, Addison Wesley, 2000.
11. M. Humphrey, F. Knabe, A. Ferrari, A. Grimshaw, "Accountability and Control of Process Creation in the Legion Metasystem", *Symposium on Network and Distributed System Security*, IEEE, 2000.
12. A. Ferrari, F. Knabe, M. Humphrey, S. Chapin, and A. Grimshaw, "A Flexible Security System for Metacomputing Environments", *High Performance Computing and Networking Europe*, 1999.
13. John T. Kohl, B. Clifford Neuman, Theodore Y. Ts'o, "The Evolution of the Kerberos Authentication Service", *EurOpen Conference*, 1991.
14. Massachusetts Institute of Technology Kerberos Team, Kerberos 5 Release 1.0.5. <http://web.mit.edu/kerberos/www/>.
15. M. A. Sirbu, John Chung-I Chuang, "Distributed Authentication in Kerberos Using Public Key Cryptography", *Symposium on Network and Distributed System Security*, IEEE, 1997.
16. M. Schunter, M. Waidner, "Architecture and Design of a Secure Electronic Marketplace", *8th Joint European Networking Conference*, pp.712.1-712.5, 1997.
17. M. Waidner, "Development of a Secure Electronic Marketplace for Europe", *ESORICS '96*, Springer, Vol. 1146, pp.1-14, Springer, 1996.
18. W. Diffie, M. E. Hellman, "New directions in cryptography", *IEEE Transactions on Information Theory*, Vol. 22, No. 6, 1976.
19. T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms", *IEEE transactions on Information Theory*, Vol. IT31, No. 4, 1985.
20. G. White and U. Pooch, "Problems with DCE Security Services", *Computer Communication Review*, Vol. 25, No. 5, 1995.
21. T. Parker, D. Pinkas, *SESAME V4 Overview*, SESAME Issue1, 1995.
22. Joris Claessens, *A Secure European System for Applications in a Multi-vendor Environment*, <https://www.cosic.esat.kuleuven.ac.be/sesame/>, 2000.

An Asymmetric Traceability Scheme for Copyright Protection without Trust Assumptions

Emmanouil Magkos^{*1}, Panayiotis Kotzanikolaou¹, and
Vassilios Chrissikopoulos²

¹ Department of Informatics, University of Piraeus, Karaoli & Dimitriou 80,
185 34 Piraeus, Greece, pkotzani@unipi.gr

² Department of Archiving and Library Studies, Ionian University, Old Palace
Corfu, 49100, Greece, vchris@ionio.gr

Abstract. Traceability schemes have been proposed as a method to establish copyright protection of broadcast information. With asymmetric traceability, the merchant cannot frame an innocent user, while no user can abuse the system without being detected. We propose an asymmetric solution for traceability, based on the very efficient symmetric scheme of Kurosawa-Desmedt [13]. We do not make any trust assumptions about the broadcasting center or other authorities. Furthermore, we establish anonymity protection for all honest users: the identity of a user is protected, until a “fingerprint” of that user is found on a pirate decoder. We make use of well-known cryptographic techniques, such as oblivious transfer, time-lock puzzles and blind signatures. Finally, we propose a cut-and-choose technique to assure the correctness of the decryption keys.

1 Introduction

With the rapid development of new IT technologies and electronic commerce, intellectual property for digital content has been an important issue. Many watermarking and fingerprinting techniques, mostly based on classical steganography, have been proposed [15]. While digital fingerprints help the content owner to identify a copyright violator (*i.e.*, a “pirate”), digital watermarks give the means to prosecute the pirate. Especially for broadcast information, such as pay-per-view TV, web broadcast of online stock quotes, online databases and CD-ROM distribution, fingerprinting techniques have been combined with cryptography to enhance identification of redistributors. In these *traceability* schemes (also known as *traitor tracing* schemes [7]), the data supplier broadcasts encrypted information, and only the authorized users are able to decrypt it by using their unique decryption keys. If an unauthorized user (pirate) get a decryption key from an authorized user (traitor) then the pirate decoder contains secret information that allows the data supplier to identify the traitor.

* Research supported by the Secreteriat of Research and Technology of Greece.

Copyright protection of digital data comes in three flavors. In the *symmetric* case (e.g., [7,3,22,2]), the merchant knows the fingerprint (e.g. marking codes or decryption keys) that is uniquely linked with the buyer. A major drawback of this approach is that if the merchant finds an illegally redistributed version of the original data, then there are no means to prove to the Court that the buyer is guilty, since it could have been the merchant himself who tried to incriminate the buyer. In the *asymmetric* case (e.g., [17,16,19]), only the buyer knows the data with the fingerprint. If, at some later time, a redistributed version of the data is found, then the merchant can identify the buyer and prove this to the Court. The *anonymous asymmetric* case (e.g., [18]) comes as a complementary protection for the buyer: Users buy information anonymously, but they can be identified in case of illegal redistribution.

The Symmetric Case. The first traceability schemes in the literature are due to Fiat and Naor [7]. However, these schemes are very inefficient: every user personal key consists of $O(k^2 \log n)$ decryption keys and the data supplier has to broadcast $O(k^4 \log n)$ ciphertexts. Recently, Kurosawa and Desmedt proposed very efficient traceability schemes [13], the K-D schemes, where each buyer has only one decryption key. They first derived lower bounds and proposed an optimum one-time scheme that satisfies these bounds. More recently, Boneh and Franklin presented a traceability scheme [2], which is provably secure and achieves *full-traceability*, i.e., there is a tracing algorithm that catches all traitors that participate in the construction of a pirate decoder. In addition, their scheme offers *black-box traceability*, i.e., the pirate decoder is not opened but only queried in order to identify the traitors. They also describe a *linear attack* against the K-D scheme, where two or more traitors are able to construct a pirate decoder that cannot be traced back to them. This attack was later addressed by Kurosawa, Burmester and Desmedt [12]. The new tracing algorithm presented in [12], makes the K-D scheme full-traceable and black-box traceable. In addition, the gap between the ciphertexts and the plaintexts in the Boneh-Franklin scheme is greater than the lower bounds achieved in the K-D scheme. As a result, and to the best of our knowledge, the most efficient traceability scheme in the literature is the K-D optimum traceability scheme.

The Asymmetric Case. Pfitzmann [16] combined the Fiat-Naor symmetric scheme with a *two-party* protocol [6] in order to turn it into an asymmetric scheme. Later, Pfitzmann and Waidner [19] got the same result by using another two-party protocol [10] and secure cryptographic commitments [5]. However, the above schemes are based on the very inefficient symmetric scheme of [7].

In [4], Kurosawa and Desmedt presented an asymmetric version for each of their two basic symmetric schemes. Asymmetry is based on the existence of some trusted entities, i.e. an arbiter or trusted agents. These entities possess the decryption keys of all the users of the system, but they are trusted not to frame any user. We believe that the trust granted to these entities is unsatisfactory: there should be a protocol, which, if executed between the broadcasting center and a

user, would establish asymmetry without the involvement of a third party. This protocol should also be secure against a misbehaving center or/and a malicious user.

Our Contribution. In this paper we turn the very efficient traceability scheme of Kurosawa-Desmedt [13] into an asymmetric scheme, without assuming the involvement of a trusted entity. For this reason we make use of a cryptographic technique called *oblivious transfer*. In addition, we propose a *cut-and-choose* technique that assures, with a non-negligible probability, the correctness of the private keys that are obviously transferred to the users of the system. Our solution can directly be embedded in the key generation procedure of the K-D scheme. Furthermore, our solution offers extra anonymity protection for the buyer, *i.e.*, the K-D scheme is turned into an *anonymous* asymmetric scheme. In order to establish anonymity, we make use of well-known cryptographic tools and techniques, namely *time-lock puzzles* and *blind signatures*.

Organization of the Paper. This paper is organized as follows. Section 2 describes the basic building blocks required for our anonymous asymmetric scheme. In Section 3, we present an anonymous asymmetric version of the K-D scheme. Section 4 concludes the paper.

2 Building Blocks

The Kurosawa-Desmedt Traceability Scheme. In [13], two traceability schemes are described. The first is an optimum one-time traceability scheme, while the second is a multiple-use scheme. While our solution, presented in Section 3, could be applied to both schemes, in this paper we deal with the optimum scheme, for simplicity reasons. A Data Supplier chooses a uniformly random polynomial $f(x) = a_0 + a_1x + \dots + a_kx^k$ over $GF(q)$ as the encryption key e_T , where q is a prime with $q > n$ for a set of n authorized users.

Key Generation. The Data Supplier gives to each authorized user u_i the personal decryption key $e_i = \langle i, f(i) \rangle$, $i = 1, 2, \dots, n$.

Encryption. The Data Supplier encrypts a session key s , as: $h = (h_0, h_1, \dots, h_k) = (s + a_0, a_1, \dots, a_k)$. Then, the Data Supplier broadcasts h to all users of the system.

Decryption. From the header h and the decryption key e_i each user u_i computes s as: $(h_0 + h_i i + \dots + h_k i^k) - f(i) = s$

Tracing. When a pirate decoder is confiscated, the pirate key e_p is exposed. If e_p contains $\langle j, f(j) \rangle$ for some user u_j , then the Data Supplier decides that user u_j is a traitor.

Non-interactive Oblivious Transfer. The notion of ($\frac{1}{2}$) Oblivious Transfer (OT) was suggested by Even, Goldreich and Lempel [9] as a generalization of Rabin’s “oblivious transfer” [20]. Imagine that Bob has two strings S_0 and S_1 . As a function of these and Alice’s public key P_A , he computes a message $OT(S_0, S_1)$ and sends it to Alice. Using her secret key x , Alice can extract from $OT(S_0, S_1)$ exactly one of the strings S_0 and S_1 , and Bob will not know which of the two Alice got. Bellare and Micali [1] presented an attractive version of the OT mechanism, where there is no need for an interaction between the sender and the receiver. We will now summarize the non-interactive oblivious transfer protocol.

Setup. Let p be a prime and g be a generator of Z_p^* . $C \in Z_p^*$ is a globally known parameter that does not have discrete logarithm modulo p . Ways of finding such numbers are described in [1].

Key Generation. Alice chooses $x \in \{0, \dots, p-2\}$ at random and sets $a_0 = g^x$ and $a_1 = C(g^x)^{-1}$. Her public key is (a_0, a_1) and her secret key is x . Correctness of Alice’s public key is achieved by verifying that $a_0 a_1 = C$.

Oblivious Transfer. Bob chooses $y_0, y_1 \in \{0, \dots, p-2\}$ at random and computes $\beta_0 = g^{y_0}, \beta_1 = g^{y_1}$. He also uses Alice’s public key to compute $\gamma_0 = a_0^{y_0}$ and $\gamma_1 = a_1^{y_1}$. Finally, he computes $r_0 = S_0 \oplus \gamma_0$ and $r_1 = S_1 \oplus \gamma_1$, and sends $OT(S_0, S_1) = (\beta_0, \beta_1, r_0, r_1)$ to Alice. On receiving β_0, β_1 Alice uses her secret key to compute $\gamma_i = \beta_i^x$. Finally, she computes $r_i \oplus \gamma_i = S_i$. According to the Diffie-Hellman assumption [8], Alice will be able to produce γ_0 or γ_1 , but never both. As a result, Alice is finally left with exactly one-out-of-two secrets and Bob does not know which secret Alice has ended with.

Time-lock Puzzles. The notion of Time-lock Puzzles was suggested by Rivest, Shamir and Wagner [21]. With time-lock puzzles, Alice can encrypt a message M so that Bob can decrypt it after a period of T seconds, with T be a real (not CPU) time period. There are several ways to implement time-lock puzzles. In [21], the message is encrypted with an appropriately large symmetric key, which in turn is encrypted in such a way that Bob can decrypt it only by performing $t = TS$ sequential squarings (i.e., each time squaring the previous result), where S is the number of squarings per second that Bob can perform. The computational problem of performing these squarings is not parallelizable in any way: having two processors is no better than having one.

Blind Signatures. Blind Signatures, suggested by Chaum [4], are the equivalent of signing carbon-paper-lined envelopes. A user seals a slip of a paper inside such an envelope, which is later signed on the outside. When the envelope is opened, the slip will bear the carbon image of the signature. In order to establish correctness in a blind signature protocol, a cut-and-choose technique can be used: Alice sends m blinded messages to Bob, and then un-blinds any $m - 1$ indicated by Bob. Bob signs the remaining message. There is a tradeoff between choosing a large m (strong correctness), and a small m (efficiency).

3 An Anonymous Asymmetric Traitor Tracing Scheme

We present an asymmetric version of the K-D scheme [13]. Essentially, the K-D key generation procedure is transformed into a 2-round protocol between any user of the system, say Alice, and the Data Supplier. We do not make any trust assumptions about the Data Supplier or another entity. The basic cryptographic primitive that we make use of is a *non-interactive oblivious transfer* of secrets (see Section 2): The Data Supplier associates Alice with two unique K-D decryption keys, *i.e.*, $S_0 = \langle i, f(i) \rangle$ and $S_1 = \langle j, f(j) \rangle$. He then obviously transfers these secrets to Alice, in such a way that Alice finds out exactly one of the two K-D decryption keys, S_0 or S_1 . The Data Supplier does not know which key Alice ended with, and Alice does not know the value of the other key.

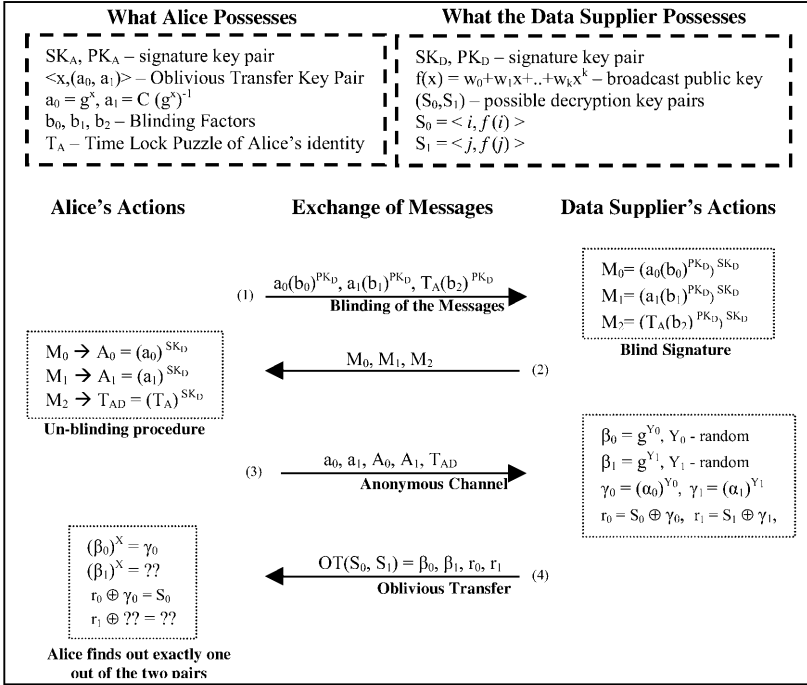


Fig. 1. An asymmetric key generation protocol for the K-D scheme

Executing the Protocol. In Step 1 (see Figure 1) Alice creates a private/public key pair $\langle x, (a_0, a_1) \rangle$ for the oblivious transfer protocol and a time-lock puzzle of her real identity, T_A . Alice blinds both (a_0, a_1) and T_A , then signs¹ the blinded messages with her signature key and sends the result to the

¹ There is a Certificate Infrastructure and users are legally bound by their signature. Mechanisms to establish non-repudiation for digitally signed messages are discussed in [23].

Data Supplier. The Data Supplier checks the correctness of the blinded messages (see also Section 2), signs them and returns them to Alice (Step 1).

In Step 2, Alice un-blinds M_0, M_1, M_2 to obtain $A_0 = a_0^{SK_D}, A_1 = a_1^{SK_D}, T_{AD} = T_A^{SK_D}$, where SK_D is the signature key of the Data Supplier. Then she sends A_0, A_1, T_{AD} and her public key (a_0, a_1) to the Data Supplier through an anonymous channel² (Step 3). The Data Supplier verifies his signature on a_0 and a_1 and uses them to prepare an instance for the oblivious transfer protocol, $OT(S_0, S_1)$, where S_0, S_1 are two K-D decryption keys. In Step 4, Alice executes $OT(S_0, S_1)$ and finds out exactly one of the two K-D keys, while the Data Supplier is not able to determine which key Alice extracted.

Tracing. When a pirate decoder is confiscated, the private decryption key e_u of an unknown user u is exposed. The Data Supplier solves the time-lock puzzle assigned with the decryption key, identifies the user u , decides that u is a traitor and presents the credentials of the user's guilt in front of a judge. Note that the new tracing algorithm of [12], which establishes full-traceability and black-box traceability, could also be used.

Security. If the Data Supplier wants to incriminate Alice, then he can construct a fake pirate decoder and randomly select one of the two keys that were obviously transferred to Alice. The Data Supplier will incriminate Alice with a success probability of $1/2$. We expect that the Data Supplier will not risk of making a false accusation, as Alice can always prove in front of a judge that her decoder contains a decryption key other than what the pirate decoder contains. In such case, the cost of the Data Supplier's false accusations will be significantly greater than the gain from a successful framing.

Remark 1. Instead of using a $(\frac{1}{2})$ oblivious transfer protocol, a $(\frac{1}{N})$ protocol could be used [14], where Alice would choose one out of N messages. The choice of N implies a trade off between correctness and efficiency. In such case, the probability of success for a false accusation would be equal to $1/N$.

We also achieve anonymity for Alice: the Data Supplier does not know the identity of Alice, when she asks for a K-D decryption key. Instead, he possesses a time-lock puzzle of Alice's identity, which can be used in case of Alice being a traitor. The use of the time-lock puzzle offers an extra protection for Alice. If the Data Supplier wants to incriminate Alice, then he must start solving very difficult computational problems, *i.e.*, the time-lock puzzles of the users that apply for a decryption key.

² There is an anonymous channel where users can send/accept messages that cannot be traced (e.g., by using traffic analysis). For example, e-mail anonymity can be established using Mixmaster re-mailers [5]. HTTP anonymity can be established by using services such as the Onion-Routing system [11].

3.1 Assuring the Correctness of the K-D Decryption Keys

In the protocol presented above, the Data Supplier is supposed to select correct K-D decryption keys as input for the oblivious transfer protocol with Alice. At the end of the protocol, the Data Supplier does not know which key Alice possesses, since he obviously transferred two different K-D decryption keys to Alice, and Alice selected exactly one of these keys. But, what happens if the Data Supplier uses two identical K-D keys as input for the oblivious transfer protocol? Or, even worse, what happens if the Data Supplier obviously transfers the same key(s) to another eligible user?

If the Data Supplier used two identical keys as input for the oblivious transfer protocol with Alice, then the Data Supplier would finally get to know Alice's decryption key, since Alice would choose one-out-of two identical keys. In such case, asymmetry would be broken and the Data Supplier could frame Alice.

On the other hand, if there is no way to assure that each instance of the oblivious transfer protocol contains two keys that are not re-used in any other instance for a different user, then there is a non-negligible probability that two legitimate users will extract the same K-D decryption key, by executing different oblivious transfers. In case of dispute, it would not be clear if an innocent user was falsely accused as a traitor or if the user is actually a traitor. Thus, traitor tracing would not be possible.

For the rest of the section, we describe *cut-and-choose* techniques that force the Data Supplier to choose correct decryption keys for each oblivious transfer. A disadvantage of these techniques is that they require some interaction between the Data Supplier and the user. Clearly, the following conditions are critical for the security of the system:

Condition A. For each user i , $i = 1..n$, the instance of the oblivious transfer protocol $OT_i(S_{i,0}, S_{i,1})$ must contain two different K-D decryption keys, *i.e.*, $S_{i,0} \neq S_{i,1}$.

Condition B. For any two different users i, j , the two instances of the oblivious transfer protocol $OT_i(S_{i,0}, S_{i,1})$ and $OT_j(S_{j,0}, S_{j,1})$ must contain different K-D decryption keys, *i.e.*, $S_{i,0} \neq S_{i,1} \neq S_{j,0} \neq S_{j,1}$.

Enforcing Condition A. We propose a cut-and-choose technique, which can be applied on the oblivious transfer protocol to ensure that for each user, the Data Supplier obviously transfers two different K-D keys, with overwhelming probability. We denote the encryption of a message m with a key X as $Enc_X(m)$.

Let $OT_i = \langle Enc_{K_i}(S_{i,0}), Enc_{K_i}(S_{i,1}) \rangle$, $i \in (0, 1)$, be two instances of the oblivious transfer protocol (see step 4 - Fig.1), where $S_{i,0}, S_{i,1}$ are four different K-D decryption keys and K_i are two encryption keys of a deterministic symmetric encryption scheme (e.g. DES keys). Moreover, let C_{K_i} be a commitment on the key K_i . For example, the commitment could be the output of a hash function (e.g. MD5) applied over each key K_i . The Data Supplier digitally signs $(OT_0, C_{K_0} || OT_1, C_{K_1})$ and sends it to Alice.

Phase 1: Checking the correctness of the encryptions. Alice randomly selects one of the two instances (say without loss of generality OT_0) and the Data Supplier reveals all secret information for this instance (*i.e.*, $y_0, y_1, \gamma_0, \gamma_1$ - same notation as in Fig.1) in order to prove that it is well constructed. Given this information, Alice retrieves both $Encr_{K_0}(S_{0,0})$ and $Encr_{K_0}(S_{0,1})$. She compares these values and if $Encr_{K_0}(S_{0,0}) \neq Encr_{K_0}(S_{0,1})$, then Alice is convinced that $S_{0,0} \neq S_{0,1}$ since the same key K_0 has been used to encrypt both K-D keys and the symmetric encryption scheme is deterministic. If the encrypted values are equal, then this means that the cleartext values $S_{0,0}$ and $S_{0,1}$ are equal too, so the Data Supplier has cheated and is reported to the judge.

Phase 2: Checking the correctness of the symmetric key. If the checks in phase 1 are correct, Alice executes the remaining instance of the oblivious transfer. In our example, this would be OT_1 . The outcome of the oblivious transfer will be either $Encr_{K_1}(S_{1,0})$ or $Encr_{K_1}(S_{1,1})$. At this time, the Data Supplier is asked to send the corresponding symmetric key K_1 to Alice. Alice checks the correctness of K_1 in two steps: first, she re-constructs the one-way commitment C_{K_1} to verify that she has been given the key that the Data Supplier had committed to. Second, she uses it to decrypt the outcome of the oblivious transfer, and see if a valid K-D key is generated. If either check fails, the Data Supplier has cheated and is reported to the judge; Otherwise, he has followed the protocol with overwhelming probability and Alice has used K_1 to obtain an official K-D key.

Analysis. A misbehaving Data Supplier has a non-negligible probability of getting caught, since he does not know *a priori* which instance of the oblivious transfer he will be asked to open in phase 1, and which instance Alice will execute in phase 2. Thus, if one of the two instances is not well constructed, then the Data Supplier's misbehavior will be revealed with probability $1/2$ for each user.

Note that the Data Supplier could try to cheat by encrypting two identical K-D keys $S = S^*$ with two different symmetric keys $K \neq K^*$. In such case $Encr_K(S) \neq Encr_{K^*}(S^*)$, while $S = S^*$, and the check in phase 1 will succeed. However, the Data Supplier has already committed to either K or K^* and cannot *a priori* determine which of the two encryptions Alice will extract from the oblivious transfer protocol. Thus, the Data Supplier's misbehavior will be revealed with probability $1/2$ for each user.

Remark 2. Because the keys are encrypted, there is no wasting of the pair of keys $S_{0,0}$ and $S_{0,1}$ that were used in the instance OT_0 : While the instance is revealed to Alice in phase 1, the Data Supplier does not send the symmetric encryption key K_0 to Alice since in that case Alice would decrypt $Encr_{K_0}(S_{0,0})$ and $Encr_{K_0}(S_{0,1})$ to obtain both keys. As a result, the Data Supplier can later use this pair of K-D keys for another user, since none of these keys has been revealed to Alice. However, the Data Supplier should re-encrypt the K-D keys with a different symmetric key.

Enforcing Condition B. In order to force the Data Supplier to use different K-D keys for different oblivious transfers, we make use of a bulletin board. Our technique can be seen as an extension of the technique used to enforce *Condition A*. We require that each instance of the oblivious transfer also contain the hash values of the K-D keys, *i.e.*:

$$OT_i = \langle C_{K_i}, \text{Encr}_{K_i}(S_{i,0}), \text{Encr}_{K_i}(S_{i,1}), \text{hash}(S_{i,0}), \text{hash}(S_{i,1}) \rangle$$

After the protocol for enforcing *Condition A* is executed (recall that in our example Alice executes OT_1), the Data Supplier publishes the pair of the hash values $\langle \text{hash}(S_{1,0}), \text{hash}(S_{1,1}) \rangle$ along with a statement that the K-D keys that correspond to these hash values have been registered to an authorized user. After executing OT_1 , Alice is left with either $S_{1,0}$ or $S_{1,1}$. Then, Alice hashes that key and checks if the result matches with the corresponding hash value that was sent to her with OT_1 . Furthermore, Alice checks the board to see if the pair of the hash values that was sent to her have been registered in the bulletin board. If not, the Data Supplier is reported to the judge. Finally, if the hash of Alice's K-D key has been registered more than one times, the Data Supplier is reported for cheating.

4 Conclusion

In this paper we presented an asymmetric traceability scheme for copyright protection of broadcast information, without the involvement of any trusted entities. For this reason we transformed the key generation protocol of the very efficient Kurosawa-Desmedt scheme [13] into an oblivious transfer protocol between the Data Supplier and the user. Each user finds out exactly one out of two unique decryption keys and the Data Supplier does not know which key the user has selected. The identity of the user is "hidden" in a time-lock puzzle that is opened in case of the user being a traitor. Our scheme is secure against a misbehaving Data Supplier, or/and a malicious user.

Applications. Our scheme, if combined with the KD scheme, could be used to identify copyright violators in applications that require broadcast encryption such as pay-per-view TV and software distribution (e.g. online databases) through the web.

References

1. Bellare, M., Micali, S.: Non-Interactive Oblivious Transfer and Applications. *Advances in Cryptology-CRYPTO '89*, LNCS 435, Springer-Verlag, 1990, pp. 544–557.
2. Boneh, D., Franklin, M.: An Efficient Public key Traitor Tracing Scheme. *Advances in Cryptology-EUROCRYPT '90*, LNCS 1666, Springer-Verlag, 1999, pp. 338–353.

3. Boneh, D., Shaw, J.: Collusion Secure Fingerprinting For Digital Data. *Advances in Cryptology—CRYPTO '95*, LNCS 963, Springer-Verlag, 1995, pp. 452–465.
4. Chaum, D.: Blind Signatures for Untraceable Payments. *Advances in Cryptology—CRYPTO '82*, Plenum Press, 1982, pp. 199–203.
5. Chaum, D.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, Vol. 24(2), 1981, pp. 84–88.
6. Chaum, D., Damgard, I., Graaf, J.: Multiparty Computations Ensuring Privacy of Each Party's Input and Correctness of the Result. *Advances in Cryptology—CRYPTO '87*, LNCS 293, Springer-Verlag, 1988, pp. 87–119.
7. Chor, B., Fiat, A., Naor, M.: Tracing Traitors. *Advances in Cryptology—CRYPTO '94*, LNCS 293, Springer-Verlag, 1994, pp. 257–270.
8. Diffie, W., Hellman, M.: New Directions in Cryptography. *IEEE Transactions on Information Theory* IT-22, November 1976, pp. 644–654.
9. Even, S., Goldreich, O., Lempel, A.: A Randomized Protocol for Signing Contracts. *Communications of the ACM*, Vol. 28, 1985, pp. 637–647.
10. Goldreich, O., Micali, S., Wigderson, A.: How to Play any Mental Game -or- a Completeness Theorem for Protocols with Honest Majority. *19th STOC*, 1987, pp. 218–229.
11. Goldschlag, D., Reed, M., Syverson, P.: Onion Routing for Anonymous and Private Communications. *Communications of the ACM*, Vol. 42(2), 1999, pp. 39–41.
12. Kurosawa, K., Burmester, M., Demedt, Y.: Proven Secure Tracing Algorithm For The Optimal KD Traitor Tracing Scheme. *DIMACS Workshop on Management of Digital Intellectual Property*, April 17–18, 2000.
13. Kurosawa, K., Demedt, Y.: Optimum Traitor Tracing. *Advances in Cryptology—EUROCRYPT '98*, LNCS 1403, Springer-Verlag, 1999, pp. 145–157.
14. Naor, M., Pinkas, B.: Oblivious Transfer and Polynomial Evaluation. *31th ACM Symposium on Theory of Computing*, ACM, 1999, pp. 245–254.
15. Peticolas, F., Anderson, R., Kuhn, M.: Information Hiding—A Survey. In *Proceedings of the IEEE, Special Issue on Protection of Multimedia Content*, IEEE Vol. 87(7), 1999, pp. 1062–1078.
16. Pfitzmann, B.: Trials of Traced Traitors. *Information Hiding Workshop*, LNCS 1174, Springer-Verlag, 1996, pp. 49–64.
17. Pfitzmann, B., Schunter, M.: Asymmetric Fingerprinting. *Advances in Cryptology—EUROCRYPT '96*, LNCS 1070, Springer-Verlag, 1996, pp. 84–95.
18. Pfitzmann, B., Waidner, M.: Anonymous Fingerprinting. *Advances in Cryptology—EUROCRYPT '97*, LNCS 1233, Springer-Verlag, 1997, pp. 88–102.
19. Pfitzmann, B., Waidner, M.: Asymmetric Fingerprinting for Larger Collusions. *ACM Conference on Computer and Communication Security*, ACM, 1997, pp. 151–160.
20. Rabin, M. O.: How to Exchange Secrets by Oblivious Transfer. *Technical Memo TR-81*, Aiken Computation Laboratory, 1981.
21. Rivest, R., Shamir, A., Wagner, D.: Time-Lock Puzzles and Timed-Released Crypto. *LCS Technical Memo MIT/LCS/TR-684*, 1996, <http://www.theory.lcs.mit.edu/~rivest/RivestShamirWagner-timelock.ps>
22. Stinson, D., Wei, R.: Combinatorial Properties and Constructions for Traceability Schemes. *SIAM Journal on Discrete Mathematics*, Vol. 11(1), 1998, pp. 41–53.
23. You, C., Zhou, J., Lam, K.: On the Efficient Implementation of Fair Non-Repudiation. *1997 IEEE Computer Security Foundations Workshop*, IEEE CS Press, 1997, pp. 126–132.

An Application Architecture for Supporting Interactive Bilateral Electronic Negotiations

Michael Rebstock

Faculty of Business Administration and Economics
Fachhochschule Darmstadt University of Applied Sciences
Darmstadt, Germany
rebstock@fh-darmstadt.de

Abstract. We develop an application architecture for interactive, bilateral, semi-structured, multi-attribute electronic negotiations in business-to-business environments. We analyze the information and the process flow for this type of negotiations. We introduce the 'Negotiation Engine' as the core component of the application and discuss its functionality. We analyze relevant application scenarios and use a reference example from the oil industry to demonstrate the practical benefit of the application.

1 Introduction

What are interactive bilateral electronic negotiations about? Like in 'bricks-and-mortar'-commerce, two market partners start these negotiations when their respective offers to buy and sell do not match in the first case and when they are willing to negotiate. In the business world, apart from auctions and tenders, nearly all everyday non-electronic negotiations are interactive bilateral negotiations.

In electronic markets theory and practice, a lot of work has been done on the 'electronification' of auctions and tenders (among many others, [9], [11], [15], [21]). Many research efforts concentrate on agent technology for electronic market transactions (e.g., [3], [8], [13], [14]). The future negotiation process in many respects might be automated. New forms of multilateral negotiations are expected to evolve, which were not known in the non-digital world [20]. For full process automation, there are some obstacles though. Agent technology today is far from being able to automate complex negotiations. And even given the technological ability to fully automate all kinds of negotiations, many users might not be ready to let go control [12]. Against this background, we expect that when it gets to the legally binding contract, many electronic negotiations will also remain *bilateral* and *interactive*. It may be desirable to analyze negotiation protocols that are well defined, in order to describe exact algorithms able to optimize negotiation results. For multi-attribute negotiations, multi-attribute utility theory can be employed and optimization algorithms can be implemented. But for the reasons mentioned, our approach here simply is to enable market partners to pursue negotiations electronically - without, at least in the first stage, using optimization algorithms.

To cope with real world contract complexity, these negotiations have to be *multi-attribute* negotiations. To be able to process the negotiation results in in-house sys

tems and thus achieve the necessary efficiency, the transaction information exchanged will have to be *structured* (while accompanying messages may be unstructured).

Some work has already been done on bilateral and multi-attribute negotiations, either conceptually (e.g., [12], [24]), or in application design and development (the Internege systems [14], IBM's market place developments [7], or Menerva [17]). Some of the applications developed originate from workflow systems and contract management systems (CrossFlow [10], DiCarta [5]). Based on these findings, we are going to investigate into some functional aspects of *interactive, bilateral, semi-structured, multi-attribute negotiations* in business-to-business electronic markets. For this purpose, we first analyze the information flow and then discuss our generic electronic negotiation application architecture. We focus on the analysis of its key component, the 'Negotiation Engine'. We explain the process flow and behavior of this component. A reference example from the oil industry is used to demonstrate the practical benefit of the application, further relevant application scenarios are analyzed.

2 Definitions

An electronic market is an application based on electronic communication services that supports the market coordination of economic activities. The legal and economic core of an electronic market transaction is a *contract* between the market partners involved. Electronic market transactions usually are conceptualized as having four phases [19]:

- *Knowledge phase* (where the relevant information concerning products, market partners etc. is gathered)
- *Intention phase* (where offers concerning supply and demand are specified by the market partners)
- *Agreement phase* (where the terms and conditions of the transaction are defined and the contract is closed)
- *Execution phase* (where the agreed-upon contract is executed and payment is made)

An *electronic negotiation application* enables the negotiation of one or more attributes of an electronic market transaction. Different negotiation protocols can be applied, depending on the number of parties involved on either side of the transaction. Though usually there are many partners on the demand and on the supply side of an EM, a specific negotiation can also be, partly or completely, carried out by only one partner on the demand side (e.g., tender processes), one partner on the supply side (e.g., auction processes), or one partner on either side (*bilateral negotiations*). In our context of bilateral negotiations in business-to-business markets, *negotiation* means a process of making and adjusting offers until a contract agreement is reached or the process is terminated without an agreement. The negotiation process is part of the agreement phase of an EM transaction, which also includes other steps like matching and scoring [20]. In this paper, we will focus on the negotiation itself and will not deal with matching and scoring functions.

3 Information Flow Analysis

In the *MultiNeg* project, we are analyzing and developing an electronic negotiation application as a functional part of an electronic market application. The electronic negotiation application thus forms a plug-in for the overall electronic market; it is conceptualized as a large-grained electronic commerce component. The electronic negotiation component exposes standard interfaces that are based on XML messages (for a general description of component-based electronic commerce applications in this sense see [8]). General electronic market functionality like catalog services or settlement services is supplied by other electronic market components and not part of the electronic negotiation component. Like the project as a whole, the *MultiNeg* electronic negotiation component focuses on bilateral, multi-attribute electronic negotiations.

Concerning the information exchange between the market partners, in contrast to structured approaches (EDI, catalog, auction) and non-structured approaches (workflow, e-mail), the electronic negotiations architecture presented here shows structured (transaction content) as well as unstructured aspects (accompanying explanatory text). We call this type of transactions *semi-structured electronic transactions*.

To define the electronic negotiation component's context, we briefly analyze the overall information flow within an electronic market that supports electronic negotiations (*Figure 1*).

Initially, offer data has to be issued by a market partner. Within our generic scenario, each of the market partners can post offers, whether buyers or sellers. On the seller side, an offer can be a specific sales offer or general catalog data. Offers issued by the buyer side can be general or specific requests for quotation. The initial offer data can be entered manually or can be imported into the electronic negotiation component via XML messages out of another electronic market component (e.g., a catalog component) or out of an in-house system (ERP system). Once an initial offer is created, a market partner can issue a counter-offer within the electronic negotiation component. The exchange of offers and counter-offers can be practiced as often as necessary. With the agreement of the parties involved, the contract is closed. The electronic negotiation component then provides both parties with the transaction data they have agreed upon, for further reference and for processing in in-house systems (ERP systems).

4 Application Scenarios and Reference Example

Application scenarios for the type of negotiation situations and the negotiation application discussed in this paper come from a variety of industries. These include, among others, process industry (chemical or pharmaceutical), the oil and gas industry, the mill industry (wood, paper), fashion industry (on the buying side: cloth, dyes; on the selling side: collections), automotive industry (pilot projects already deployed include *Matsushita's* Internet-based supplier network which covers, among other functionality, price negotiations [22]), high-tech industry, software service industry (a pilot project from this industry is reported in [1]) and other service industry sectors. In all cases, transactions show a medium degree of structure and standardization combined

with a medium degree of transaction frequency. The transactional complexity involved makes full automation by agents difficult to achieve, but still allows electronic systems support - medium transaction frequency leaves generic electronic system support being still economically attractive.

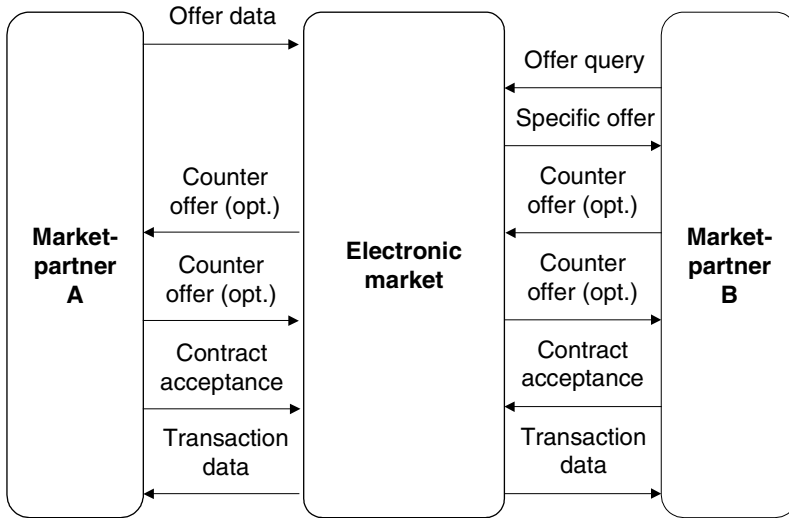


Fig. 1. Electronic Market Information Flow

Not all industries are feasible though. Where transactions are highly structured, standardized, or show high transaction rates (like in the commodities industry, consumer products and retail industry; or with transactions on exchanges), full process automation is easier to achieve and economically attractive. On the other hand, where transaction structure is very low, or where very low transaction rates are common (like for industrial installations manufacturers or for large projects in the constructing industry), other forms of electronic transactions (e.g., tenders) seem to be more appropriate.

To illustrate the application situations we are dealing with, we use a reference scenario from the oil industry. Our scenario is based on a real-world project; for this paper, the application situation has been simplified. For our example, let *B* be a purchasing agent at *B Corp.*, a local marine services company in one of the world's major ports, offering its products and services to shipping companies at port. *B Corp.* resells the products of a major multinational oil company, to which it is tied by reseller contracts. *S* is a salesperson in the marine products division of *S Corp.*, the multinational oil company. The marine products division's market is the marine business worldwide. They offer the supply of marine fuels and marine and industrial lubricants to resellers at all the world's major ports.

For their transactions, *B*'s and *S*'s prices are not completely fixed in advance but subject to actual market prices and the specific order situation (e.g., quantities, qualities required, delivery date, reseller status). Actual market prices are constantly changing (spot market prices). Additional charges can be added depending on the

specific order. Product specifications can be individually determined and prices are also dependent on this individual product specification. Different qualities ('grades') of the product as well as delivery date and its type of packaging can have to be negotiated. Some items may be replaced by substitutes at another price if necessary. All adjustments can become necessary in the course of the electronic negotiation process. Thus the flexible and interactive negotiation of a variety of attributes, like product types, qualities, quantities, delivery dates, and price elements based on the respective reseller involved, has to be supported by our electronic negotiation application.

5 Generic Application Architecture

Generally an electronic negotiation application can be implemented in different ways: It can be implemented on the *buyer's side*, on the *seller's side*, or it can be implemented as a third-party, *intermediary* application. In this paper, we will not detail these alternatives. In the context analyzed here, the implementation alternatives do not influence the functionality of the application; in all cases, market partners can use its functionality in the same way.

Within our application architecture, we find three major components (*Figure 2*):

- *Communication Engine*: This component handles incoming and outgoing messages. It includes authentication and encryption functionality. It also supplies workflow functionality, which is used to manage the negotiation processes. All communications are based on XML messages.
- *Negotiation Engine*: This component enables electronic negotiations and manages transaction content. Negotiations are based on partner messages received by the Communication Engine and reference the application's business object framework.
- *Transaction Organizer*: This component supports the management and 'meta-negotiation' of the application's business object framework. It is discussed in more detail in another paper [18].

This paper focuses on the core negotiation component, the 'Negotiation Engine', and the underlying negotiation process.

6 Negotiation Process

The general flow of the negotiation process is represented in *Figure 3*. As we focus on the flow of control instead of event-triggers, we use an activity diagram instead of state machines. As we are here only interested in the negotiation process itself, we do not include other agreement phase activities like matching or scoring.

The process starts after an initial offer has been placed by one of the market partners and the offer has been perceived by another market partner. This can be achieved either through direct offer entry or through browsing and selecting offers on the market place using an electronic catalog or similar electronic market place functionality. In our reference example, *B* places a RFQ if he needs to restock certain fuels or lubricants. He may place an *ad hoc*-RFQ if a particular vessel needs a specific grade or type of packaging. *S* offers price (depending on, among others, actual spot market prices), grade, type of packaging and delivery date. Both *B* and *S* can browse the offers posted in the system.

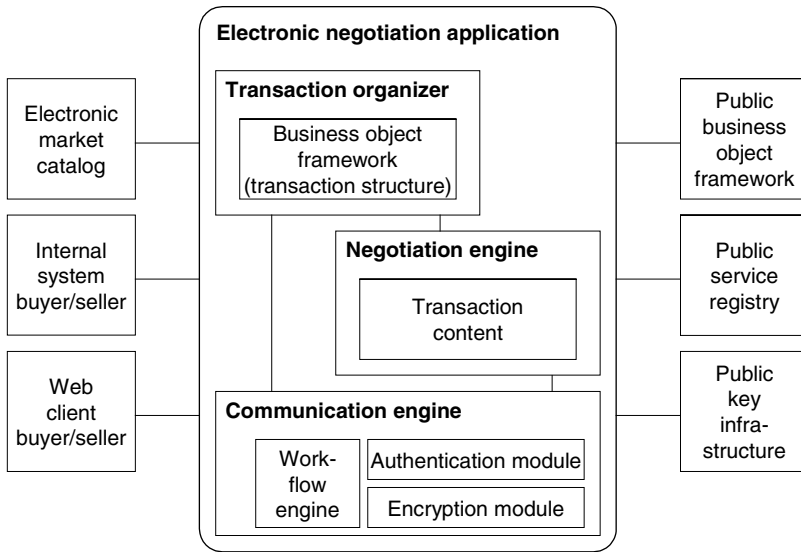


Fig. 2. Electronic Negotiation Application Architecture

Based on the offer issued, the market partners can start a negotiation. If a negotiation is requested by one of the market partners, the other partner has to agree. In this case, a negotiation transaction is initialized, which means that initial transaction content and an initial transaction definition are created, based on the initial offer. The market partners can then start to negotiate, modifying the attribute values of the initial offer. Each attribute value change is submitted to the market partner for confirmation.

In our example, *S* could suggest an alternative grade or type of packaging, if she cannot supply the lubricants in the exact way *B* requested them. *B* might want to lower the price because otherwise he cannot resell the products at a reasonable margin. *S* might respond that it is not possible to lower the price, but that another lubricant product with similar qualities, but different additives could be sold at the price suggested. *B* might not want this product as he knows that the vessel's machine cannot be operated with the alternative lubricant offered. He might insist in negotiating the price. *S* might then offer a delivery date in seven days time, as she expects spot market prices to drop within this period. As *B* knows the vessel is still on sea and will not arrive at port before next week, he might accept *S*'s offer.

In order to expand the possible agreement space of their negotiation, the market partners could come to the decision that they have to add a further attribute to their negotiation. In this case, the negotiation of the transaction content is suspended and the transaction definition (i.e. its information structure as part of the application's business object framework) is negotiated instead. We will not detail the process and benefits of this feature here, as mentioned above, this is done in another paper [18].

If the market partners agree on all aspects of the respective transaction, the contract is closed. Closing the contract electronically includes all activities and functionality necessary to install a complete, consistent and legally binding contract. Within this paper, we will not go into the details of the necessary technical functions such as

authentication, encryption or translation. The complexity of this function is directly linked to the amount of trust already established by other sources such as framework contracts, long-term business relations or trusted third parties present within the relationship of the parties involved. In our reference example, as reseller contracts do already exist, a legal framework generating trust is already established. The requirements for this function therefore are comparatively low.

To achieve truly *interactive* electronic negotiations, the functional characteristics of electronic market systems have to be merged with those of workflow systems. In general, workflow systems are related to intra-organizational process flows. In our case, we are dealing with *inter-organizational* process flows. Still the usual criteria and characteristics of workflow systems - such as process models, status concept, responsibilities, event management or exception handling - are applicable to the workflow functionality of an electronic negotiation application as well. (These services, provided by the Communication Engine within our application architecture, will be specified in a future paper. Already, significant work has been done on this topic [10].) We expect the communication flow within our electronic negotiation application normally to be *asynchronous* - the market partners have different time slots available to deal with the negotiation. Reasons for this are manifold: conflicting other communications, meetings or travel time, different office hours, or different time zones. The *MultiNeg* electronic negotiation application component serves as a platform for the exchange of structured messages as well as additional, unstructured information. When a market partner posts a counter-offer within the application, she can post some explanatory text together with her structured offer data (simply achieved by supplying an additional text element in the respective XML message.)

Within the process of a negotiation, a market partner has to ensure that a new suggestion is feasible in terms of his business needs and capabilities. The electronic negotiation application therefore has to supply an (again XML-message based) interface that allows to check whether the suggested values of the contract attributes are feasible. This information is retrieved from in-house systems. Often these will be ERP systems, in other cases, legacy systems or database applications. These systems check whether new prices still allow for a margin, they check whether new delivery dates are possible or whether different grades or qualities can be delivered or produced. For each attribute in the business object framework, a service can be defined that checks the feasibility of a suggested value.

In our example, *B* needs to check whether *S*'s suggestion of a later delivery date is feasible, therefore he has to check the vessel's arrival date at port in his order processing system. *S* needs to check what alternative product grades she can offer *B* at a lower price. She checks her stock and calculates her minimum cost using her sales system. (In our case, both *B*'s and *S*'s systems are functional modules of their respective ERP systems.)

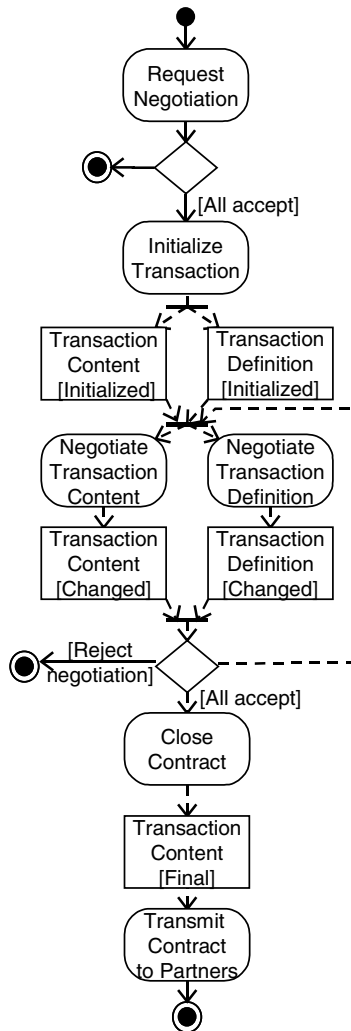


Fig. 3. Electronic Negotiation Activity Diagram

7 Conclusion and Future Research

We have maintained that *interactive, bilateral, semi-structured, multi-attribute negotiations* will be an important part of electronic market functionality. We have discussed the flow of the negotiation process and developed a generic application architecture for this type of negotiation. We consider our approach a modest, pragmatic one. We will develop a prototype that covers real-world complexity of business-to-business transactions and for transaction optimization rely on human actors. After completing the *MultiNeg* prototype, we will evaluate its usability and efficiency in

test and real world environments. With this data about the simple interactive negotiation process, we hope to see more clearly where and to what extend the addition of optimization algorithms and negotiation support agents can enhance electronic negotiation applications.

Many other questions are open to future research. Establishing efficient interfaces between electronic negotiation applications and in-house systems to our mind is crucial for a wide acceptance and universal usability of electronic negotiation systems. The same holds true for the ability to flexibly deal with diverse business object frameworks. Developing and deploying flexible protocols for business content messages seems to continue being one of the most vital tasks in electronic markets research and development.

References

1. Addis, M., Allen, P., Surridge, M. (2000): Negotiating for Software Services, in: Tjoa, A.M., Wagner, R.R., Al-Zobaidie, A. (eds.): Proceedings of the 11th International Workshop on Database and Expert Systems Applications, Los Alamitos, CA, 1039-1043
2. Bichler, M., Werthner, H. (2000): A Classification Framework of Multidimensional, Multi-Unit Procurement Negotiations, in: Tjoa, A.M., Wagner, R.R., Al-Zobaidie, A. (eds.): Proceedings of the 11th International Workshop on Database and Expert Systems Applications, Los Alamitos, CA, 1003-1009
3. Chkaiban, G., Sonderby, M. (2000): AATP: Auction Agent Transfer Protocol, in: EM - Electronic Markets 10, No. 2, 94-101
4. Damian, D.E.H., Shaw, M.L.G., Gaines, B.R. (2000): A Study of Requirements Negotiations in Virtual Project Teams, in: Hansen, H.-R., Bichler, M., Mahrer, H. (eds.): Proceedings of the 8th European Conference on Information Systems (ECIS2000), Volume 2, Vienna, 937-944
5. DiCarta (2001): System Preview, <http://www.dicarta.com>, 24.01.2001
6. ebXML Technical Architecture Project Team (2001): ebXML Technical Architecture Specification v1.0, http://www.ebxml.org/specdrafts/ebXML_TA_v1.0.pdf, 15.01.2001
7. IBM (2000): Virtual Market Place (ViMP), http://www.zurich.ibm.com/activities/csc/csc_e-business-vimp.html, 02.11.2000
8. Jennings, N.R., Wooldridge, M. (1998): Applications of Intelligent Agents, in: Jennings, N.R., Wooldridge, M. (eds.): Agent Technology: Foundations, Applications, and Markets, Berlin, Heidelberg, New York etc.
9. Klein, S., O'Keefe, R.M. (1999): The Impact of the Web on Auctions, in: International Journal of Electronic Commerce 3, No. 3
10. Koetsier, M., Grefen, P., Vonk, J. (2000): Contracts for Cross-Organizational Workflow Management, in: Bauknecht, K., Madria, S.K., Pernul, G. (eds.): Electronic Commerce and Web Technologies, Berlin, Heidelberg, New York etc., 110-121
11. Koppius, O.R., Kumar, M., van Heck, E. (2000): Electronic Multidimensional Auctions and the Role of Information Feedback, in: Hansen, H.-R., Bichler, M., Mahrer, H. (eds.): Proceedings of the 8th European Conference on Information Systems (ECIS2000), Volume 1, Vienna, 461-468
12. Liang, T., Dong, H. (2000): Effect of Bargaining in Electronic Commerce, in: International Journal of Electronic Commerce 4, No. 3, 23-43
13. Limthanmaphon, B., Zhang, Y., Zhang, Z. (2000): An Agent-Based Negotiation Model Supporting Transactions in Electronic Commerce, in: Tjoa, A.M., Wagner, R.R., Al-Zobaidie, A. (eds.): Proceedings of the 11th International Workshop on Database and Expert Systems Applications, Los Alamitos, CA, 440-444

14. Lo, G., Kersten, G.E. (1999): Negotiation in Electronic Commerce: Integrating Negotiation Support and Software Agent Technologies. Interneg Report INR03/99, <http://interneg.org/interneg/research/papers/1999/03.pdf>, 20.12.1999
15. Lopes, F., Mamede, N., Novais, A.Q. et al. (2000): Towards a Generic Negotiation Model for Intentional Agents, in: Tjoa, A.M., Wagner, R.R., Al-Zobaidie, A. (eds.): Proceedings of the 11th International Workshop on Database and Expert Systems Applications, Los Alamitos, CA, 433-439
16. Microsoft Corporation (2000): BizTalk Framework 2.0: Document and Message Specification, <http://www.microsoft.com/biztalk/techinfo/BizTalkFramework20.doc>, 02.01.2001
17. Menerva (2000): Enabling the Whole Deal, <http://www.menerva.com>, 11.12.2000
18. Rebstock, M. (2001): Efficiency and Flexibility of Multi-Attribute Negotiations - The Role of Business Object Frameworks, in: Proceedings of the 12th International Workshop on Database and Expert Systems Applications, Los Alamitos, CA
19. Schmid, B., Lindemann, M. (1997): Elemente eines Referenzmodells Elektronischer Märkte. Report-No. IM HSG/CEM/44, v1.0, University of St. Gallen
20. Ströbel, M. (2000): On Auctions as the Negotiation Paradigm of Electronic Markets, in: EM - Electronic Markets 10, No. 1, 39-44
21. Teich, J., Wallenius, H., Wallenius, J. et al. (2000): An Internet-Based Procedure for Reverse Auctions - Combining Aspects of Auctions and Negotiations, in: Tjoa, A.M., Wagner, R.R., Al-Zobaidie, A. (eds.): Proceedings of the 11th International Workshop on Database and Expert Systems Applications, Los Alamitos, CA, 1010-1014
22. The Economist (2000): Business-to-business in Japan. No room in the nest, in: The Economist 355, 15. Apr. 2000
23. UDDI (2000): UDDI Technical White Paper, <http://www.uddi.org>, 12.09.2000
24. Wedekind, H. (2000): On Specifying Contract Negotiations, in: Hansen, H.-R., Bichler, M., Mahrer, H. (eds.): Proceedings of the 8th European Conference on Information Systems (ECIS2000), Volume 1, Vienna, 23-30

Strategies for Software Agent Based Multiple Issue Negotiations

Dominik Deschner, Florian Lang, and Freimut Bodendorf

University of Erlangen Nuremberg Information Systems II,
Lange Gasse 20
D-90403 Nuremberg, Germany

{bodendorf | deschner | lang}@wi2.wiso.uni-erlangen.de

Abstract. This paper summarizes the results of a research project that yielded a prototype for agent-based multiple issue negotiations along with a package of strategies those agents use to accomplish their goals. Whereas many simple negotiation protocols for software agents have already been developed, multiple issue negotiations are seldom addressed. Here, a bilateral negotiation protocol is outlined and a model allowing autonomous software agents to balance out their diverging interests is presented. The model outlines strategies for the agents to compromise on issues based on utility calculations. To calculate its next move, an agent does not only rely on user preferences (importance of issues, attitude towards risk, ...) but also considers information from its dynamic surroundings, e.g. the current market situation. The agents also get to know and recognize each other, strategically applying their experiences from negotiations formerly completed or aborted. To ensure applicability, the algorithms designed for modeling the agents' behavior rest upon certain pragmatic assumptions like bounded rationality, limited information and self-interest.

1 Introduction

Negotiation is a process of joint decision making and an instrument for resolving conflicts. Parties with opposing interests try to find mutually favorable contracts by exchanging compromise proposals. Successful negotiation in complex environments like B2B-markets is just as difficult and unforgiving as it is essential and valuable. A lot of hard and soft factors like the opponent's explicit and implicit behavior, current market situation, own goals, time left and any other potentially valuable input determining the negotiation setting have to be considered. Deriving suitable negotiation strategies from this data is a challenging task, even for negotiators with the cognitive skills of a human being.

The approach described here is to automate negotiations by designing software agents with decision making skills, allowing self-interested agents to reach mutually acceptable deals on business transactions. Based on an inter-agent negotiation protocol negotiation strategies have been developed with functions that derive suitable actions

(offer, agree, ...) from context data (market situation, opponent's behavior, ...). The system is called *SettleBot*.

There have been many projects on agent-based negotiation before (see [1], [3], [4], [8]). *SettleBot* is extending these approaches by

1. providing a model of a negotiation platform populated by agents
2. dealing with multiple-issue negotiations. The contract to be settled can have any number of variables, which are all negotiated in parallel. *SettleBot*'s agents negotiate about 20 flexible attributes at a time.
3. modeling completely autonomous agents on both sides of the bargain table. After an agent has been authorized to settle a certain transaction, all further activities (e.g. generating an offer) are performed without human assistance.
4. adopting realistic assumptions. The agent's decision functions cope with the conditions of a competitive environment. Agents are neither properly informed about the negotiation mechanism itself (unlike game theory) nor about their opponent's options and strategy. The agents have to rely on careful assessments of the situation, thereby evaluating their next shot, which in almost every case results in may-be-successful-but-far-from-optimal behavior.

2 Basic Concept

2.1 System Architecture

The multi agent system consists of several buying and selling agents, an information agent and a blackboard.

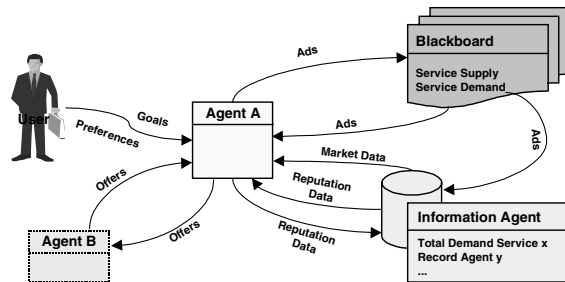


Fig. 1. Inter-Agent Communication

The agents use the blackboard (BB) for posting offers (ads). The information agent (IA) monitors the blackboard and exchanged messages collecting information about agreements, breakups and current demands and offers. Using this information the IA generates information about the World State which negotiating agents use as input to their decision functions (see fig. 1).

2.2 Negotiation Protocol

„Negotiation is a decision process in which two or more agents make individual decisions - formulate compromise proposals. The proposals are communicated to other agents. Upon receiving a counter proposal, a new proposal is determined. The process continues until either an agreement or a deadlock is reached” ([6], S. 449). In fig. 3 this process is modeled by a sequence of actions. First, an agent generates an ad, indicating its intention (buy, sell), the transaction object (e.g. rental car) and a negotiation space for all issues (e.g. free miles between 100 and 500)¹. Other agents retrieve the ad and evaluate it in terms of compatibility to their own intentions. If there is a common negotiation space for all issues, an agent starts a negotiation thread by submitting an initial offer. This thread is an iterative process of generating offers, evaluating offers and deciding whether to proceed, ending with either an agreement or termination (see fig. 2).

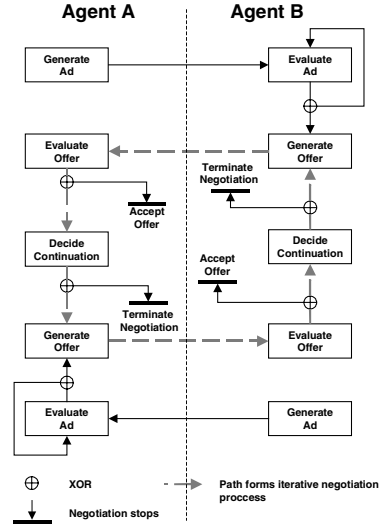


Fig. 2. Negotiation protocol

2.3 Negotiation Object

To allow agents to refer to attributes of a negotiated object the object's attributes and their valid values have to be specified. A rental car might have attributes like class, free mileage, price, etc. All attributes form negotiation issues. Generating offers and calculating utilities is dependent on data type (number, date, string, boole) and scale (nominal, ordinal, metric) of the attribute. To allow communication, agents have to comply with a common ontology. Since communication in *SettleBot* is based on exchanging XML-Messages the ontology can be defined by a XML-Schema, e.g. defining <PRICE> or <PAYMENT_DUE> as common elements.

2.4 Contract Utility and User Preferences

The goal of a Self-Interested Agent (SIA) is to maximize the utility it attains from a contract. Here it is assumed that this utility is determined by the conformity of a con-

¹ Knowledge of these values cannot be exploited because an agents decision making is based on differing current reservation values, which are calculated depending on negotiation strategy and world state (see below).

tract to user goals. The user defines a reservation value (RV), which yields a utility of 0 and an aspiration value (AV) with a utility of 100 for any attribute. Considering x_i as the value of an attribute i , the function $v_i(x_i)$ assigns a utility v_i to that value.²

The utility of a contract concerning an object with n attributes can be calculated by adding up the weighted utilities of the attribute values i.e.

$$v_{\text{abs}}(x) = \sum_{i=1}^n g_i v_i(x_i) = g_1 v_1(x_1) + g_2 v_2(x_2) + \dots + g_n v_n(x_n).$$

The weights g_i depend on the attributes' relative importance and add up to 1. To keep preference elicitation simple, a utility function is applied that does not model compound effects and interdependencies between attributes. Also for usability, linear utility functions are assumed, simply asking the user to specify AV_i , RV_i and g_i . Additionally, the user provides the deadline T_{max} at which settlement must be reached, the risk attitude ω ($\omega=1$ is risk-neutral) and the maximum duration of a negotiation t_{crit} . T_0 indicates the instantiation of a negotiating agent. Since $t_{\text{crit}} \ll T_{\text{max}}$ T_0 many negotiations can be conducted during an agent's life span.

3 Negotiation Activities and Decisions

3.1 Generating an Initial Offer

Since the negotiation protocol is iterative and several offers can be exchanged, a rational agent will try to gain high utility by initially submitting a maximum demand. If an initial offer is accepted, the submitting agent can safely assume to have wasted utility that would have been gained by submitting a more demanding initial offer. To minimize the risk of wasting utility and to maximize the residual negotiation space, a rational agent submits its aspiration values AV_i .

3.2 Evaluating an Offer

An offers' acceptability is judged by comparing its utility with a threshold value. Ideally, this threshold is determined such that an offer is not accepted if in succeeding negotiations a better result can be attained. Since this is unknown to the agent, the determination of the threshold value is an optimization problem with incomplete information. A high threshold value increases the risk of a breakup, a low threshold value may waste utility. The determination of a threshold value is a matter of negotiation strategy. For now it is assumed that such a threshold value v_{goal} exists.

² The slope of these functions is determined e.g. by the user's willingness to take risks. For examples of utility functions (e.g. logarithm function etc.) see ([7], p. 71 and p. 118)).

3.3 Deciding whether to Continue the Negotiation

If the threshold value has not been exceeded by the offer, the agent decides whether to proceed or not. If an agent B has not yet generated an offer, i.e. it has received and rejected the initial offer $x^{A \rightarrow B, t_0}$ (offer x , submitted by agent A to agent B at starting time t_0), a rational B will definitely continue the negotiation with its own initial offer. Agent B has a low but non-zero probability to attain the maximum utility by settling the contract $x = (AV_1^B, AV_2^B, \dots, AV_n^B)$.

In case both agents have already submitted at least one offer, the termination of the negotiation is an option. An agent B terminates negotiation if the critical duration of a negotiation has been exceeded ($t > t_0 + t_{crit}^B$) without reaching an agreement (see [3]) and agent A is making no more concessions, i.e. submits an offer $x^{A \rightarrow B, t}$ which yields the same or less utility than the previous offer $x^{A \rightarrow B, t-1}$.

3.4 Modifying an Offer

If both negotiating agents have already submitted one or more offers which have been rejected by the peer and agent B has decided to continue the negotiation, B generates a subsequent offer by modifying its previous offer $x^{B \rightarrow A, t-1}$.

The decision by the peer to accept an offer depends solely upon the quality of the submitted offers. The modification strategy is therefore suitable if it considers the interests of both parties. For a modification, a function $\pi \rightarrow x$ is necessary, which maps the current World State π , i.e. the strategically relevant data (strategy factors like time, market situation etc.) to a modified offer. Assuming that x_i represents the deviation of a modified offer from the initial offer $x_i^{t_0}$ and this value depends upon strategy factors, a modified offer for attribute i at time t can be calculated by

$$x_i^t(\pi) = x_i^{t_0} + \gamma_i(\pi) \cdot x_i(\pi).$$

In order to be able to consider different modification strategies this function has to be generalized by replacing x_i by $\gamma_i(\pi) \cdot x_i(\pi)$. γ_i represents a general modification function with a range of $[0;1]$, γ_i represents the modification space in attribute units. If an agent is willing to modify the value of an attribute i within the whole negotiation space then $\gamma_i = AV_i - RV_i$.

With this modification model it is possible to formulate modification functions and strategies $\gamma_i(\pi)$ regardless of the data type and the scale of the attribute. $\gamma_i = 0$ means no modification of the initial offer, whereas $\gamma_i = 1$ means the whole modification space is utilized, i.e. the maximum compromise for this attribute is reached. By mapping the strategy factors to a modification strategy, the agents are able to adapt to any World State thus improving their gains in the negotiation.

4 Example Strategy Model

SettleBot assumes incomplete information and limited computational resources. Thus, negotiating agents can only handle a limited view of the relevant World State to derive appropriate behavior. Such aspects of the World State, which can be measured by an agent and used to calculate proper behavior are called strategy factors. In this section methods are outlined for mapping strategy factors and user preferences to a modification function.

4.1 Strategy Factors

The strategy factors allow the agents to evaluate the current negotiation setting and to calculate an appropriate negotiation stance. The negotiation stance of an agent is split into its negotiation power and its attitude towards another agent.

An agent's current negotiation power can be derived from information publicly available from the IA. In our model, the IA calculates and publishes any agent's general reputation, average transaction volume and thread-specific market power. The higher those values, the more powerful the agent. A rational agent A with a bad reputation acts more defensive. As A does not know B's decision functions it does not know for sure whether B will try to exploit or even detect A's drawback. If it does not, defensive behavior would mean losing utility. However, A will rationally rely on the assumption that B behaves rational to a minimum degree, therefore negotiating aggressively towards an agent with bad reputation, claiming more utility from deals with such an agent. To improve its probability of reaching agreement, A will make more concessions than B and reduce its utility threshold appropriately. Thus, a powerful B has a high probability of prevailing in the negotiation, because a rational A attaches importance to a reliable transaction (good reputation of A), to future utility (high average transaction volume of A), and to an agreement with A if B is short on alternative transaction partners (market power of A).

To calculate the strategy factors π_{market}^A , π_{rep}^A and π_{vol}^A by the IA applicable methods have been developed (see [2]). An even market situation, a neutral reputation and an average transaction volume each lead to a respective value of 1. Deviations increase or decrease this value. Experiments have shown that the standardization of all strategy factors to the scale $\{\frac{1}{2}, \frac{3}{4}, 1, \frac{5}{4}, \frac{3}{2}\}$ is suitable. An agent A calculates its negotiation power by

$$^A(\pi) = \pi_{market}^A \cdot \pi_{rep}^A \cdot \pi_{vol}^A.$$

$^A > 1$ indicates high negotiation power and should lead to a more aggressive, demanding strategy of A, whereas $^A < 1$ suggests defensive behavior.

A rational agent will compare its power to the power of its peer. If A acts aggressively due to $^A = 1.4$, it may be successful negotiating with a defensive agent B. A's strategy might fail if it negotiates with a more powerful agent B with $^B = 1.8$. In

calculating its negotiation stance, agent A therefore has to consider B's relevant factors. This is calculated by the attitude

The attitude of A towards B results from the reputation of B, its average transaction volume and a "liking" of A towards B. The market power is not considered here since the negotiation partners represent opposite sides of the market (supplier and buyer) and their market power is interdependent. If A considers its own market power, implicitly the market power of all its potential peers is considered.

Reputation and transaction volume of the negotiating agents can be obtained from the IA. The liking $\pi_{likes}^{A(B)}$ is calculated by A from previous experiences with B (for a method to calculate the liking see [2]). The agent now calculates its attitude by

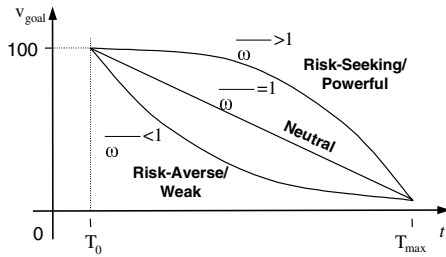
$$\pi^{A(B)} = \pi_{likes}^{A(B)} \pi_{rep}^B \pi_{vol}^B$$

These strategy factors can be weighted according to user preferences if e.g. transaction volume is more important than reputation.

4.2 From Strategy Factors to Behavior

4.2.1 Dynamic Utility Threshold

The utility threshold v_{goal} determines the minimum utility a contract must yield to be acceptable for an agent. Not only received offers are measured against this threshold, it also determines the boundaries for concessions, stating minimum utility of the offers an agent generates. Therefore, the threshold is the utility an agent claims, derived from the agent's negotiation stance. Here, a relevant strategy factor is the time left until the deadline T_{max} is reached. Agents with a lot of time left can risk a breakup because they can expect to conduct further negotiations. As T_{max} comes closer, an agent's threshold utility is continually decreasing until it becomes 0 at time T_{max} , which makes $v_{goal}(\pi)$ a decay function.



$$v_{goal}^A(\pi) = 100 \frac{T_{max}^A - t}{T_{max}^A - T_0^A} \frac{A}{A}$$

Fig. 3. Decay functions determining an agent's utility threshold

The slope of $v_{\text{goal}}(\pi)$ is determined by α , β and ω (see fig. 3). The higher the negotiation power of A the higher the threshold utility. Positive (high) attitude leads to a low threshold utility. High $v_{\text{goal}}(\pi)$ makes higher probability of a breakup. Therefore, the threshold utility must also be determined by the user's risk attitude ω . A risk-averse agent will reduce its threshold utility even if it has a strong negotiation stance.

4.2.2 Dynamic Modification Space

Based on the dynamic utility threshold, the agents calculate current modification spaces for every attribute ensuring that the utility of the generated offers does not drop below v_{goal}^A . Depending on the dynamic threshold utility current reservation values $RV_i(t)$ are calculated for all attributes, satisfying the condition

$$\sum_{i=1}^n g_i v_i(RV_i(t_0)) = v_{\text{goal}}(t_0)$$

The reservation utility of an agent B at time t for attribute i results in a current modification space

$$m_i(t) = RV_i^B(t) - AV_i^B$$

If an agent sticks to the $m_i(t)$, it never falls short of its threshold utility. To calculate the current reservation value two methods have been developed. One handles all attributes of the negotiation issue in the same way, shifting the reservation values toward the aspiration values until the threshold utility is reached. The other method distinguishes with respect to the relevance of the attributes. The modification space of important attributes is increased only if the deadline is coming much closer.

4.2.3 Concessions

A concession is a basic pattern within the process of creating modified offers. By conceding the opposing party shall be convinced to accept a proposed contract. To develop a model for concession making which describes the modification of an offer, a modification function $m_i(\pi)$ with respect to the modification space m_i is necessary. Only functions which do not allow to revoke concessions are considered, because after a critical negotiation time is reached, rational agents break up a negotiation if no further increase in utility is attained (see section 3.3). Since agents do not know each other's critical negotiation time, revoking a compromise always has to be avoided in order to continue a negotiation. Therefore, all modification functions increase in a monotonic way.

A negotiation among two agents A and B takes at least $\min(t_{\text{crit}}^A; t_{\text{crit}}^B)$ units of time, before one of the agents breaks up. The agents do not know whose critical negotiation duration exceeds the other. Since they can not determine whether the opponent's concessions come "early" or "late", they have to stick to their own critical negotiation duration and assume the same time span for their counterpart. In this case an agent A

submits its “last” offer within a negotiation started at time t_0 at time $t_0 + t_{crit}^A$. This offer is represented by the current reservation values $RV_i(t)$.

Jennings provides a set of concession patterns where agents increase their bid aggressively, neutrally or defensively ([3], see also Hartig’s work [5]). The agents in SettleBot assign an appropriate concession pattern to every attribute with respect to the strategy factors. The strategy is differentiated into a global and a attribute specific component. A generally defensive or aggressive stance of an agent is represented by ω_{glob}^A , derived from the same strategy factors that determine the decay function’s slope. This global component is calculated for an agent A by

$$\omega_{glob}^A = \frac{\omega^A}{\omega^B}.$$

The attribute-specific component ω_i considers the relevance of every attribute in calculating the concession strategy. A rational negotiating agent first concedes on less relevant attributes. This component is calculated by

$$\omega_i = (g_i \cdot n)^2$$

Multiplying the strategy factors ω_{glob}^A and ω_i (both are symmetric to 1) results in a concession strategy for attribute i . The modification functions are represented by

$$x_i^B(\pi) = \frac{t - t_0}{t_{krit}} \omega_{glob}^B(\pi) \omega_i(\pi) \left(\frac{B}{B(A)} \right) \omega^B (g_i^B \cdot n)^2 (RV_i^B(t_0) - AV_i^B)$$

The diagram illustrates the formula for calculating compromises, with callouts for each term:

- $x_i^B(\pi)$: Initial offer claiming maximum utility
- $\frac{t - t_0}{t_{krit}}$: Increasing compromise in negotiation process
- $\omega_{glob}^B(\pi)$: Negotiation stance based on reputation, liking, transaction volume and market power of both sides
- $\omega_i(\pi)$: Attitude towards risk
- $\left(\frac{B}{B(A)} \right)$: Implementation of own goals
- ω^B : Implementation of own goals
- $(g_i^B \cdot n)^2$: Implementation of own goals
- $(RV_i^B(t_0) - AV_i^B)$: Implementation of own goals

Fig. 4. Formula to calculate compromises

Combining the functions to calculate the threshold utility, the time dependent modification space and the function to calculate concessions on a certain attribute, a basic model for generating offers can be derived as depicted in fig. 4

4.3 Other Strategies

The strategy model outlined so far does not consider the goals of other agents. Therefore, several extensions of this basic strategy have been developed and applied to the basic model, e.g. a trade-off strategy, a fairness strategy derived from the tit-for-that strategy (see [9]), etc (see [2]). These strategies consider the goals of the opposite party and can be combined.

The trade-off strategy, for example assumes that an offer of A is more acceptable to B the closer it is to B's previous offer. Thus, after calculating several offers yielding the same utility to A the closest one is submitted.

Another extension applies to market rules. The negotiation protocol and strategies outlined so far can easily be applied on negotiations aiming at trust building. The coalition formation strategy (see [2]) allows agents to combine their powers by negotiating a common preference structure and then acting as one agent.

5 Conclusion

SettleBot has been prototypically implemented as a stand alone application using Borland Delphi. Several panels are available to parameterize the agents (e.g. set the aspiration values for the attributes) and to simulate certain World States. Two agents negotiate a contract by exchanging XML messages.

SettleBot can be adapted to any goods or services with multiple attributes. As has been experimentally proven, the negotiating agents agree upon mutually acceptable contracts with respect to user preferences. The outlined example strategies allow agents to selfishly exploit available information about the relevant World State.

Lack of available information is a major obstacle both for the agent and the user, who needs to determine a complex combination of parameters, often no more than guessing what is best. A user could learn to improve her tweaking skills by strictly observing an agent's behavior, but keeping user effort minimal must be an overruling design principle. Thus, further research is necessary to develop a learning mechanism to have the agents tweak themselves and converge towards an optimal strategy, most likely using reinforcement learning (see [9]).

References

1. Chavez, A., Maes, P.: Kasbah: An Agent Marketplace for Buying and Selling Goods, in: Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, London 1996, pp. 75.
2. Deschner, D., Lang, F.: Verhandlungsstrategien koordinierender Softwareagenten, Arbeitsbericht des Lehrstuhls Wirtschaftsinformatik II, Universität Erlangen-Nürnberg Nr. 2 (2001) (in preparation).
3. Faratin, P., Sierra, C., Jennings, N.: Negotiation Decision Functions for Autonomous Agents, in: International Journal of Robotics and Autonomous Systems 24 (1997) 3-4, pp. 159.
4. Guttman, R.: Merchant Differentiation through Integrative Negotiation in Agent-mediated Electronic Commerce, Cambridge 1998
5. Hartig, W.: Modernes Verhandeln, Heidelberg 1995
6. Kersten, G., Cray, D.: Perspectives on Representation and Analysis of Negotiation: Towards Cognitive Support Systems, in: Group Decision and Negotiation 5 (1995), pp. 433.
7. Mag, W.: Grundzüge der Entscheidungstheorie, München 1990
8. Oliver, J., On Automated Negotiation and Electronic Commerce, <http://opim.wharton.upenn.edu/~oliver27/dissertation/diss.zip>, 14.5.2001
9. Sandholm, T., Crites, R.: Multiagent Reinforcement Learning in the Iterated Prisoner's Dilemma, in: Biosystems 37 (1995), pp. 147.

Automatic Construction of Online Catalog Topologies

Wing-Kin Sung², David Yang¹ * **, Siu-Ming Yiu¹, Wai-Shing Ho¹,
David Cheung^{1,2}, and Tak-Wah Lam¹

¹ Department of Computer Science and Information Systems
The University of Hong Kong, Hong Kong
{dyang, smyi, wsho, dcheung, twlam}@csis.hku.hk

² E-business Technology Institute
The University of Hong Kong, Hong Kong
{wksung, dcheung}@eti.hku.hk

Abstract. The organization of a web site is important to help users get the most out of the site. Designing such an organization, however, is a complicated problem. Traditionally, this design is mainly done by hand. To what extent this can be automated is a challenging problem. Recently, there have been investigations on how to reorganize an existing web site based on some criteria. But none of them has addressed the problem of organizing a web site automatically *from scratch*. In this paper, we attempt to tackle this problem by restricting the domain to online catalog organization.

We model an online catalog organization as a decision tree structure and propose a metric, based on the *popularity* of products and the relative *importance* of product attribute values, to evaluate the quality of a catalog organization. The problem is then formulated as a decision tree construction problem. Although traditional decision tree algorithms, such as C4.5, can be used to generate online catalog organization, the catalog constructed is generally not good based on our metric. An efficient greedy algorithm (GENCAT) is thus developed and the experimental results show that GENCAT produces better catalog organizations based on our metric.

1 Introduction

The web site organization problem is a well known problem in the literature. The Araneus project [6] tries to manage a web site in database style. The Strudel project [11] provides a query language and a HTML-template language for the user to specify the web site's structure and the visual presentation, respectively. However, the linkages between pages are still specified by man and there is no scientific measurement on the goodness of the organization of a web site.

* David Yang is on leave from the Department of Mathematics and Computer Science, St. Joseph's University, Philadelphia; yang@sju.edu.

** Part of this work was developed at the IBM Centre for Advanced Studies in Toronto.

To what extent this complicated task of web organization can be automated is a challenging problem. Recently, there has been some progress along this direction. Perkowitx and Etzioni [7] supplement an existing organization by constructing index pages for related pages automatically. These related web pages are identified by mining the visitors' logs. Green [4], on the other hand, adds some cross links between pages that are likely to be related by studying the contents of the pages. While these works enhance the existing structure, they do not consider the quality of the existing structure. Garofalakis et al. [2] address this quality issue using *page popularity*. They reorganize the existing structure locally by swapping children and parent pages if the child page is more popular. However, this may not be true as there may be some semantic relationships among the pages that would make swapping pages inappropriate. Also, none of these approaches has addressed the problem of organizing a web site automatically *from scratch*. While consultation with users is of course essential, we suggest that it is possible to do this in conjunction with automatic methods.

In this paper, we attempt to tackle this problem by restricting the domain to online catalogs. An online catalog is an organization of a set of product pages through which users access their required product information. The number of product pages in an online catalog is usually high and the organization of the catalog is crucial to the success of an e-commerce web site, this specific domain serves as a good starting point for studying the more general problem.

A good online catalog organization should facilitate visitors locate *popular* products *easily* and *efficiently*. According to our knowledge, there does not exist any quantitative method to evaluate a catalog tree. We, then, propose a metric to evaluate the quality of an online catalog organization. The metric takes into account two important factors. One is the *popularity* of each product so that fewer "clicks" are needed to get to more popular products. The other is how likely a visitor can get to the right product by following the links provided by the catalog. In deciding which link to follow, visitors usually need to answer a question related to an attribute of the product, if the visitors do not care the attribute being asked, they may follow a wrong link. So questions on more important attributes should be asked earlier. Visitors can then have a higher chance to get to the right products. To ensure the proposed metric is feasible, we also develop a quantitative framework using a well-established marketing methodology, called *conjoint analysis* [5], to capture reliable measures of the relative importance of attributes. In general, based on the proposed metric, traditional decision tree algorithms, such as C4.5, do not give us a good solution because those algorithms do not consider the importance of the product attribute values. An efficient greedy algorithm, GENCAT, is developed. Experimental results show that GENCAT does give better catalog organizations under our metric.

In the next section, this automatic online catalog construction problem will be presented. Section 3 will discuss the metric, the greedy algorithm, and a brief analysis of the proposed solution. Discussion and conclusion will be given in Section 4.

2 Problem Description

2.1 Model for Online Catalog Design

When visitors try to locate product information from an online catalog, either they will make use of the search engine or they will follow the structure of the catalog going from one web page to another. Since the vocabulary used by the visitors may not match that used by the site or the visitors may not know exactly what they are looking for, search engine does not always produce satisfactory results. The organization of the catalog is especially important in these cases.

Although a real online catalog may have cross links between product pages which are usually characterized by “related items”, the underlying structure of a catalog is basically a tree. The root page (or the main page) represents the entire set of products, and the various choices to follow represented by the parent-child links indicate subsets of products. In other words, an online catalog can be regarded as a topology of the set of products. When visitors navigate through an online catalog, they essentially traverse a tree to find the product that interests them. If we ignore all those cross-links (and backward links), the product information pages are usually at the leaves of the tree. We, therefore, model a catalog structure as a tree with leaves corresponding to the product pages and the inner nodes corresponding to *navigational* pages. By navigational, we mean that the primary purpose of these pages is to guide the user towards the appropriate product pages.

Each navigational page is implicitly representing a subset of product pages. The links provided in the navigational page will further classify the corresponding subset into smaller subsets. The decision of which link to follow usually depends on the answer to a question which is related to one of the properties (attributes) of the products that interest the user.

To simplify the discussion, we make the following assumptions on product pages. Consider a set of product pages P .

1. All product pages in P are characterized by the same set of non-null attributes A .
2. Each product page can be uniquely identified by its set of corresponding attribute values.

Table 1 shows an example of 5 mobile phones which are characterized by 3 attributes: size, access wap, and call waiting. Note that we will regard a mobile of dimension $5 \times 10 \times 2$ as large and dimension $3 \times 5 \times 1$ as small.

An online catalog organization is formulated as a *catalog tree* as follows.

Definition of Catalog Tree. Given a set of product pages P , a catalog tree for P is a tree $T = (V, E)$ where V is the set of nodes and E is the set of edges such that

1. Every product page in P is mapped to a unique leaf node in V .
2. Each internal node in V is labeled by an attribute in A .

3. For each edge $e = (u, v)$ where u is the parent, if u is labeled by an attribute a , then the edge e must be labeled by a valid attribute value of a .
4. For each product page p , if the attributes, a_1, a_2, \dots, a_i with corresponding attribute values, x_1, x_2, \dots, x_i , appears on the path from the root to p , then p can be uniquely identified by the attribute set, a_1, a_2, \dots, a_i with corresponding attribute values, x_1, x_2, \dots, x_i .

Figure 1 shows three different catalog trees for the set of product pages described in Table 1. In order to know which catalog tree is better, we need some quantitative measurements. In the next subsection, we will propose the metrics to measure the quality of the catalog trees.

Table 1. Characteristics of 5 different mobile phones and part-worths of attribute values in the mobile phone example

ID	size (dimension in cm)	access wap	call waiting	popularity	attribute	value	part-worths	unimportance
A	$5 \times 10 \times 2$ (big)	no	no	25	size	big	0	6
B	$5 \times 10 \times 2$ (big)	no	yes	20	size	small	4	2
C	$3 \times 5 \times 1$ (small)	no	yes	20	access wap	yes	5	1
D	$3 \times 5 \times 1$ (small)	yes	no	50	access wap	no	0	6
E	$3 \times 5 \times 1$ (small)	yes	yes	100	call waiting	yes	2	4
					call waiting	no	0	6

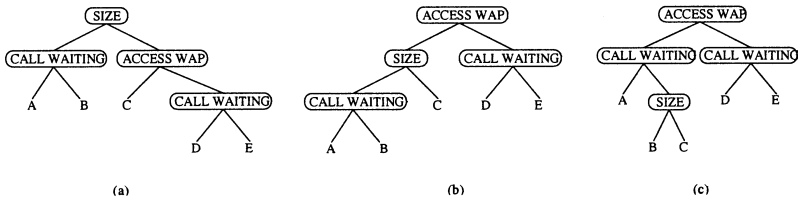


Fig. 1. Different catalog trees for the same set of products

2.2 Metric

As shown in the mobile phone example, for the same set of products, there are more than one possible catalog tree. The question of which one is the best is not easy to answer. According to our knowledge, there are no quantitative measures proposed to evaluate the quality of a catalog tree. Basically, a good catalog tree should facilitate users locate *popular* product information *efficiently* and *easily*. In this subsection, based on this idea, we try to define a reasonable metric which allows us to evaluate how good a catalog tree is.

To quantify this measurement idea, first of all, we need to have a quantitative measure for the popularity of product pages. We can approximate it by past sales

figures or number of accesses to those products. To address the efficiency issue, one approach is to minimize the number of “clicks” for the visitors to get the popular pages. This is the rationale behind $f_1(T)$, the average weighted depth metric. However, $f_1(T)$ does not address the easiness issue. When visitors browse through the catalog tree level by level, they need to pick a link in each level by answering an abstract question, “Which value of attribute a interests you the most?”. If the attribute values do not appear to be interesting to the visitor, it may not be easy for the visitor to locate the right product page. Among the attribute values which describe a particular product, some of them are important or key features while the others are not. A good catalog tree should avoid using unimportant attribute values in the navigational pages.

The importance of an attribute value can be assigned by the designer of the online catalog. However, it is not desirable and may be quite subjective. Instead, we borrow a concept, called *part-worth* from the marketing research. Basically, part-worth of an attribute value measures contribution of this particular attribute value to the overall utility of a product [3]. In other words, it reflects the relative importance of that attribute value for the product. Intuitively, the higher the part-worth of an attribute value, the more people will think that this attribute value is important. Assuming that we have obtained the part-worths of the attribute values, we will show a better metric which also takes these into account.

Let P be a set of products and A be the set of attributes used to characterize P . For every attribute $a \in A$, let V_a be the set of attribute values of a . Observe that each product $p \in P$ can be described using an attribute value vector $(x_{a_1}, x_{a_2}, \dots)$ where $x_{a_i} \in V_{a_i}$ for every attribute $a_i \in A$. Among the attribute values in the vector, some of them are critical (important) characteristic of p while the other are non-critical (unimportant). Our aim is to build a catalog tree T for P (i.e. P is the set of leaves of T) which tries to achieve two things. Let I_p be the set of attribute values appearing along the path from the root of T to p for any product page p .

- We tries to reduce the depth of every product in T , especially for the popular products.
- For each product p , we tries to minimize the number of unimportant attribute values in I_p .

To achieve the above two things, we proposes the metric, $f_2(T)$, which tries to measure the average weighted unimportance of a catalog tree.

Definition of the Average Weighted Unimportance Metric. We define a measurement $unimport_v$, for every attribute value v , such that $unimport_v$ has a large value if v is unimportant. Then, we define the metric as

$$f_2(T) = \frac{\sum_{p \in P} pop_p (\sum_{v \in I_p} unimport_v)}{\sum_{p \in P} pop_p}.$$

Intuitively, for any product $p \in P$, if $\sum_{v \in I_p} \text{unimport}_v$ is large, then p may have a high depth in T or the number of unimportant attribute values in I_p is large. So, reducing $\sum_{v \in I_p} \text{unimport}_v$ captures the above two criteria.

What remains is to define the measurement unimport_v for every attribute value v . There are many different ways to achieve this. Here, we try to define unimport_v based on the part-worth of v , that is u_v . It is nature to deduce that the larger the u_v , the more important the v is. Hence, we denote $\text{unimport}_v = m - u_v$ where m is a constant which is larger than the part-worth of all the attribute values.

Table 1 shows the values of part-worths and unimportances for the mobile phone example. Based on this table, Figure 1(b) gives the best organization under metric $f_2(T)$.

2.3 Problem Definition

Formally speaking, the automatic online catalog construction problem can be defined as follows. Let A be the set of attributes that characterize the set of product pages P .

Input: A set of product pages P and the metric $f_2(T)$

Output: A catalog tree T for P which minimizes $f_2(T)$.

As state in the following lemma, solving this problem is NP-hard. In the next section, we propose a greedy algorithm to solve this problem.

Lemma 1. *The automatic online catalog construction problem is NP-hard.*

Proof. Please see [9] if you are interested at the proof.

3 Our Proposed Solution

Since a catalog tree bears a resemblance to a decision tree, traditional decision tree construction algorithms, such as C4.5 [8], can be used to build a catalog tree. However, the catalog constructed is generally not good based on our metric because they did not consider the information gained through importance measurement. In this section, we will present the greedy algorithm, GENCAT, for constructing a local optimal catalog tree under the proposed metric. The running time of the algorithm will then be analyzed and the performance of the algorithm will be compared with those of other decision tree construction algorithms.

3.1 The Greedy Algorithm - GENCAT

The algorithm consists of two steps. The first step generates an initial catalog tree T for the product pages P . The next step will restructure this tree by considering different choices of attributes at each node in a depth-first manner.

The output of the algorithm will then be a local optimal catalog tree with respect to the metric, $f_2(T)$ [9]. The following gives the details of these steps.

We start by building an initial catalog tree. Different initial trees can be used. One way to generate the initial tree is that we start building it by arbitrarily choosing an attribute at the root. The product pages will be partitioned into different subsets according to the corresponding attribute values. A child node will be created for each subset. For each child node with more than one product page, we arbitrarily pick another attribute and continue the partitioning process until every leaf node contains a single product page.

After getting the initial tree T , the second step, which is the core of the algorithm, tries to restructure T to optimize the value of $f_2(T)$. Starting from the root, we do the restructuring in a depth-first manner. For each node v , we consider every possible attribute, a . Then, the subtree rooted at v is reorganized using a as the attribute for the node v . This reorganization will modify the structure of the subtree based on *the original structure of T* . This restructuring will be shown to be always feasible using a procedure called Rebuild(). Among all these possible alternatives, we choose the one that gives the best value of $f_2(T)$. When the algorithm stops, we end up with a local optimal catalog tree with respect to the metric $f_2(T)$.

GENCAT

INPUT: the set of product pages P and the metric $f_2(T)$

OUTPUT: a locally optimal catalog tree T for P with respect to $f_2(T)$

1. Build a decision tree T which consists of all the product pages;
2. $R = \text{Improve}(T, r, f)$ where r is the root of T ;
3. Return R ;

Improve(T, v, f)

1. For every attribute a , let $T_a = \text{Rebuild}(T, v, a)$;
2. Among all the T_a , let R be the T_a such that $f_2(T_a)$ is minimized;
3. For every non-leaf child u of v , let $R = \text{Improve}(R, u, f)$;
4. Return R ;

Fig. 2. The GENCAT algorithm for automatic catalog organization

The procedure, $\text{Rebuild}(T, v, a)$, will restructure the subtree of T rooted at v using attribute a in node v . The details are as follows.

1. Change the attribute in node v of T to a .
2. Partition the product pages in the leaves of the subtree rooted at v into groups G_1, G_2, \dots, G_k according to the corresponding attribute values of a
3. For each group G_i , let S_i be a tree constructed from T as follows
 - a) all the product pages not in G_i are removed; and
 - b) all the inner nodes which have one child are removed.
4. Replace all the subtrees which are attached to v by S_1, S_2, \dots, S_k .

¹ In fact, the algorithm is quite general in the sense that if we substitute the metric $f_2(T)$ by another appropriately defined metric, the algorithm still works

Figure 3 shows an example how $\text{Rebuild}(T, w, A_5)$ restructures the subtree rooted at w using attribute A_5 to replace A_2 in node w . The second step shows the result of partitioning leaf pages rooted at w according to the attribute values of A_5 . The third step shows the result of restructuring the subtree rooted at w based on the original structure of T . The final output is obtained by removing internal nodes which have only one child (for example, the nodes using attributes A_3 and A_4).

Figure 2 shows the algorithm for GENCAT. We can make use of GENCAT even if not all attributes apply to every product. We can deal with this by first limiting ourselves to the attributes which do apply to all products. Next the remaining products within each of the subtrees can be grouped by the groups of products which each have the same more specialized attributes. 2

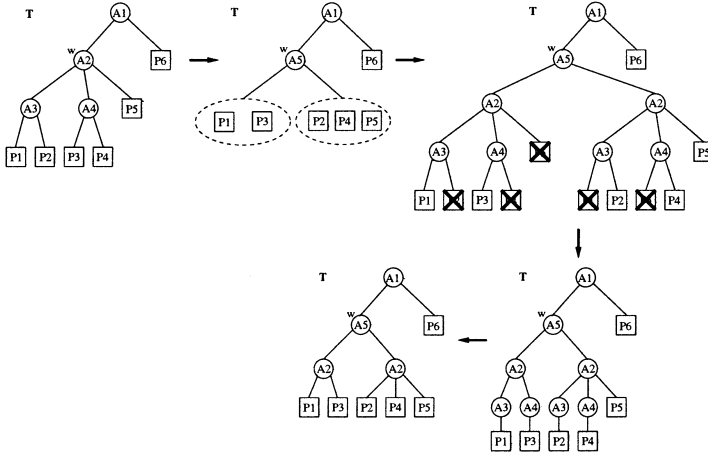


Fig. 3. An example which demonstrates $\text{Rebuild}(T, w, A_5)$

GENCAT can also help if the catalog tree must start with a standard classification. For example, if a clothing retailer who wants to pre-divide the products into groups, say clothing for women, men, and infants. There can be shirts or blouses in each category with the same set of attribute values. In such a case, the site designer would divide up the products into the appropriate subsets, then run GENCAT on each subset.

3.2 Performance Analysis

In this section, we will discuss the performance of the algorithm. It is shown that the time complexity of GENCAT is polynomial in the number of product

² Thanks to Vijay Iyengar for pointing this out.

pages and the number of attributes per product page. Then, we will compare the performance of our algorithm with those of other decision tree construction algorithms.

Time Complexity

Lemma 2. *GENCAT requires $O(|A||P|^2)$ time where A is the set of attributes and P is the set of products in the catalog.*

Proof. Please refer to [10] for the detailed proof.

Performance Tests on Synthetic Data. The performance tests were divided into two parts. We first compared the results from GENCAT with those from C4.5. Then we tested the performance of GENCAT over different input data sets.

Comparison between C4.5 and GENCAT. For the comparison between C4.5 and GENCAT, we tried 50 data sets and each data set has 500 pages, 20 attributes and there are at most 4 possible values per attribute. Table 2 shows the means and standard deviations of the experiment results. It shows that although C4.5 could give similar result as GENCAT in terms of $f_1(T)$, the trees from C4.5 gives a much higher $f_2(T)$ values than those from GENCAT. This means that C4.5 does not consider the unimportance of the attribute values, and tries to ask the users more questions on unimportant attributes and forces users to choose between unimportant attribute values in the internal nodes of the catalog trees.

Table 2. Comparison between GENCAT and C4.5

	C4.5	GENCAT
	mean std dev	mean std dev
$f_1(T)$	4.845 0.043	4.638 0.084
$f_2(T)$	25.56 0.95	14.88 1.10

Performance of GENCAT. We also tested two other aspects of the performance, performance ratio and computation time, of GENCAT over different input data sets. The performance ratio is defined as the ratio of $f_2(T)$ value of the catalog tree produced by GENCAT to that by an optimal tree generator which uses a brute force optimal algorithm with dynamic programming. We found that the increase of number of product pages n_p or number of attributes n_a would make GENCAT slightly less effective. However, GENCAT still has an average ratio less than 1.1 in our test cases, which means that our trees are just 10% deeper than the optimal ones. We only performed tests for up to 100 pages or 12 attributes because it is nearly infeasible (the running time for a single test case is more than 2 hours) to find an optimal tree for test cases with more product pages or attributes.

Figure 4 shows that GENCAT has a good scalability over the parameters. The running time was only slightly quadratic (nearly linear) to n_p or n_a . For a data set with 1500 product pages and 20 attributes per product page, the time for running GENCAT was only 20 minutes.

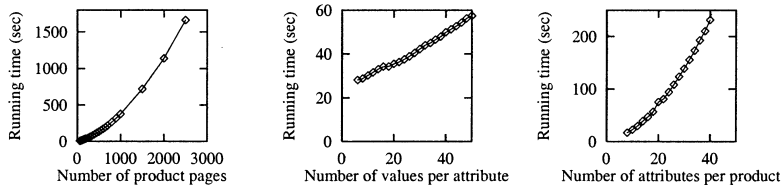


Fig. 4. Effect of different parameters on running time

4 Discussion and Conclusion

This paper introduces the problem of automatic online catalog construction which is formulated as a decision tree construction problem. A metric is proposed to evaluate the quality of a catalog organization. An efficient greedy algorithm, GENCAT, is developed. Experiments show that GENCAT does produce better catalog organizations.

References

1. M. Fernandez, D. Florescu, J. Kang, A. Levy, and D. Suciu. STRUDEL - a Web site management System. In *ACM SIGMOD International Conf. on Management of Data (Exhibits Program)*, 1997.
2. John Garofalakis, Panagiotis Kappos, and Dimitris Mouloukos. Web site optimization using page popularity. *IEEE Internet Computing*, 3(4):22-29, 1999.
3. Paul E. Green, Douglas S. Tull, and Gerald Albaum. *Research for Marketing Decisions*. Prentice Hall, 5th edition, 1988.
4. Stephen J Green. Automated link generation: can we do better than term repetition? *Computer Networks and ISDN Systems*, 30(1-7):75-87, 1998.
5. Joy P. Guilford. *Psychometric Methods*. McGraw-Hill, 2nd edition, 1954.
6. G. Mecca, P. Atzeni, A. Masci, P. Merialdo, and G. Sindoni. The ARANEUS Web-Base Management System. In *ACM SIGMOD International Conf. on Management of Data (Exhibits Program)*, 1998.
7. Mike Perkowitz and Oren Etzioni. Adaptive web sites: automatically synthesizing web pages. In *AAAI*, pages 727-732, 1998.
8. J. Ross Quinlan. *C4.5: programs for machine learning*. Morgan Kaufman, San Mateo, CA, 1993.
9. Wing-Kin Sung, David Yang, Siu-Ming Yiu, David Cheung, Wai-Shing Ho, and Tak-Wah Lam. On the complexity of an online catalog tree construction problem. manuscript.
10. Wing-Kin Sung, David Yang, Siu-Ming Yiu, David Cheung, Wai-Shing Ho, Tak-Wah Lam, and Sau-Dan Lee. Automatic construction of online catalog topologies. submitted to *IEEE Transactions on Man, Cybernetics and Systems*.

A Two-Layered Integration Approach for Product Information in B2B E-commerce

Borys Omelayenko and Dieter Fensel

Division of Mathematics and Computer Science
Vrije Universiteit, De Boelelaan 1081, 1081 hv
Amsterdam, The Netherlands
borys@cs.vu.nl, dieter@cs.vu.nl

Abstract. Electronic B2B marketplaces bring together many online suppliers and buyers, each of which can potentially use his own format to represent the products in his product catalog. The marketplaces have to perform non-trivial mappings of these catalogs. In this paper, we analyze the problems which occur during the integration, taking several leading XML-based standards as an example. We advocate a three-layer product integration framework to resolve the difficulties in overcoming these problems with a direct one-layer integration. In this paper, we focus on the first two layers: the XML-based *syntax layer* and the *data models layer* expressed in RDF. The approach operates in three main steps. First, we create an RDF data model from the XML catalog, which eliminates all syntactical peculiarities of the catalog. Second, the catalog is translated from the source model to the RDF model of the target catalog. Finally, the transformation from RDF to XML restores all syntactical regulations required by the target catalog format. The approach is suitable for inter-operation with higher-level document and workflow ontologies.

1 Introduction

The World Wide Web has drastically changed the on-line availability of information and the amount of information exchanged electronically. The web has revolutionized personal information access and knowledge management in large organizations (cf. [7]). In addition, it has begun to change the commercial relationships between suppliers and customers. The [17] estimates for the business-to-consumer (B2C) area range between \$4 billion to \$14 billion on-line sales in the US for 2000, which is approximately 1% of the overall sales figures. This is still a small fraction of the overall business figures, but we can expect its fast growth given the fact that the number of Internet users grew from 100 to 300 million between 1997 and 2000. Similar estimates have been made for the business-to-business (B2B) area. Forecasts for the dollar value of B2B EC in the US range between \$600 billion to \$2.8 trillion for 2003 (cf. [17]). Currently, a large fraction of B2B transactions is still realized by traditional non-Internet networks, such as those conducted over EDI systems. In this

traditional paradigm, direct one-to-one connections and mappings are programmed based on standards, such as EDIFACT¹. However, this traditional paradigm does not employ the full power of electronic commerce and it will soon be replaced by the Internet and Web-based transaction types.

Electronic marketplaces for Business-to-Business (B2B) electronic commerce bring together many online suppliers and buyers, which participate in business interactions (cf. [16] for an overview of the field). Internet and web-based electronic commerce provide a much higher level of *flexibility* and *openness*, which helps to optimize business relationships. According to the U.S. Department of Commerce [17] estimates, there were around 800 B2B marketplaces in early 2000. Other studies estimate around 10,000 B2B marketplaces in the very near future. These marketplaces provide completely new opportunities for their clients:

- A supplier, linked to a marketplace is automatically linked to a large number of potential customers, instead of implementing one-to-one links to each supplier.
- A supplier or a customer can choose between a large number of potential business partners and can optimize his business relationships.

Concisely, B2B marketplaces are a middleware, which helps the customers to contact a large number of potential clients without running into the problem of implementing a large number of communication channels. However, preventing the customers from the bottleneck of the exponential growth in the number of implemented business connections becomes a serious problem for B2B marketplaces. They contend with the problem of heterogeneity in the *product*, *catalogue*, and *document* description standards of their customers. Efficient management of different description styles becomes a key task for these marketplaces. In addition, a number of serious mapping problems need to be solved in order to make the B2B area working. These mapping tasks arise at several levels² (cf. [16] for an overview):

- Different standards for describing products, or content standards (e.g., UN/SPSC³ versus ecl@ss⁴).
- Different standards for describing the structure of the product catalogues, which contain the links to the content standards (e.g., Ariba⁵ versus CommerceOne⁶).
- Different standards for describing exchanged business documents, such as purchase orders (e.g., XML Common Business Library xCBL⁷ versus Commerce XML cXML⁸).

The first type of mappings mainly involves the real-world *semantics* of the information exchanged. They appear due to users who describe the same products in

¹ www.unece.org/trade/untdid/welcome.htm

² Not to mention the problem of mapping XML-based standards for updating product catalogues.

³ www.unspsc.org

⁴ www.eclass.de

⁵ www.ariba.com

⁶ www.commerceone.com/solutions/business/content.html

⁷ www.xcbl.org

⁸ www.cxml.org

different ways and with different product classification schemes. The second and the third types of mappings arise in connection with the *syntactical structure* of the information exchanged. The first mapping is usually provided by a content management solution provider. The second and the third mappings are usually provided by the B2B marketplace itself. From the technical point of view, the second and the third tasks are a kind of a document integration task, where the catalogs and exchanged documents must be translated and linked together. Most of the participants use XML to encode their documents, and a number of XML-based product description standards have been developed (cf. [6], [12]). There are around ten leading document standards for the B2B area existing now, but this number may increase further in the near future. Non-XML standards, such as EDIFACT or ISO STEP [9] are still in use. However, the corresponding XML encoding for them has been already developed and is now in the process of standardization. Hence, non-XML integration is unlikely to draw major interest in the future.

Some of these XML standards are compared by Li [12], who discusses seven different product description standards used in e-commerce, their complexity and potential integration problems. We will discuss this topic more extensively, proposing our architecture able to solve the integration problems highlighted in [12].

The XML transformation language XSL-T [3], together with the correspondent expression language XPath [4], provides the means to translate various XML documents. However, attempts to define the integration rules directly in XSL-T have revealed several important problems that make development of real-life integration rules very complicated [15]. Aside from the fact that XSL-T provides a low level of service for defining such mappings, the single-layer integration is a conceptual mismatch that poses serious obstacles to a direct mapping approach. Complex rules need to be defined to extract the semantic information from various syntactical styles, which translate the information at a semantic level, and which represent this information in a different syntax. Such rules are difficult to write and maintain; their re-usage is very limited. We have sought the solution to these problems in a layered approach for the catalog integration task. We distinguish between different sub-tasks in the overall mapping process, which enable identification of simple and re-usable rule patterns. Complex transformations are reached through concatenation of a number of simple transformation rules.

In this paper, we discuss a layered model for business document integration and take the integration of address descriptions as a running example discussed in Section 2. Section 3 sketches the problems that arise in direct single-layered catalog integration. Section 4 introduces the three-layered model for the catalogs which consist of a *Syntax layer*, a *Data Models layer*, and an *Ontology layer*. Section 5 discusses the two-layered catalog integration approach. The paper ends with conclusions and future research directions.

2 The Running Example of XML Catalogs

We use the problem of *address* integration as a running example throughout the paper. An address is a simple business concept. It occurs very frequently in e-commerce and is an important part of any B2B mediation system. Unlike most of products, the structure of an address and the meaning of its components are understandable to everybody, which makes the explanation clear. At the same time, the alignment of various address representations of an address provides all major types of problems, which can appear in the product integration task.

The first standard analyzed in the paper is **xCBL** 3.0 developed by Commerce One⁹ Inc. It provides a comprehensive set of standardized XML document formats, allowing buyers, suppliers, and service providers to integrate their existing systems into the electronic marketplaces [5]. The second standard is **cXML** 1.0 developed by a large consortium of companies including Ariba and Microsoft. cXML has been proposed for a similar purpose as xCBL, and it also targets document integration for the B2B mediation task. The DTDs for the address representation according to the above-mentioned two standards are presented in Fig. 1 (a)-(b) correspondingly.

```
<!ELEMENT OrganizationAddress ((AddressType)?, (ExternalAddressID), (POBox)?, (Street)?,
(HouseNumber)?, (StreetSupplement1)?, (StreetSupplement2)?, (PostalCode)?, (City), (Country),
(Region)?, (District)?, (County)?, (TradingPartnerTimezone)?)>
```

```
<!ELEMENT AddressType ((AddressTypeCoded), (AddressTypeCodedOther)?)>
```

(a) xCBL

```
<!ELEMENT PostalAddress (DeliverTo*, Street+, City, State?, PostalCode?, Country)>
```

(b) cXML

Fig. 1. The DTDs for an address

The representations of the same concept, the address, differ in each catalog. Conceptually equal document properties (e.g. denoting a street name) can be encoded with XML elements of different names; XML elements with the same names can have different meanings (especially this refers to the elements with unclear meaning, like *Type*), and ordering which is important in XML. Finally, the descriptions may have different granularity levels as required by the domain area, and provide or omit additional details. An example of the same address encoded with xCBL and cXML is shown in Fig. 2.

3 The Problems of the Single-Layered Integration

In the simplest case, we can integrate two XML catalogs directly by defining a set of XSL-T rules, as discussed in [15]. The rules directly translate each element or attribute of the first catalog into an appropriate XML element of the second catalog. See Fig. 3 (XSL-T rules, which directly map the cXML address to the xCBL format).

⁹ www.commerceone.com

<pre> <OrganizationAddress> <ExternalAddressID>001</ExternalAddressID> <POBox/> <Street>De Boelelaan</Street> <HouseNumber>1081a</HouseNumber> <StreetSupplement1/> <StreetSupplement2/> <PostalCode>1081 hv</PostalCode> <City>Amsterdam</City> <Country>Netherlands</Country> <Region>North Holland</Region> <District/> <County/> <TradingPartnerTimezone/> </OrganizationAddress> </pre> <p>(a) xCBL</p>	<pre> <Address> <Name xml:lang="en">VU</Name> <PostalAddress name="VU"> <DeliverTo>B. Omelayenko</DeliverTo> <Street>De Boelelaan, 1081a</Street> <City>Amsterdam</City> <State/> <PostalCode>1081 hv</PostalCode> <Country isoCountryCode="NL">Netherlands</Country> </PostalAddress> </Address> </pre> <p>(b) cXML</p>
--	---

Fig. 2. The xCBL and cXML catalog examples

This approach mixes several independent tasks in a single XSL-T rule as shown in **Fig. 3**

- Aligning different terminologies, e.g. mapping the xCBL element **OrganizationAddress** and the cXML element **PostalAddress**.
- Aligning the granularity level of the representations and performing necessary attribute splits with XPath expressions. For example, the **Street** cXML element, which actually refers to an address line with the street name and house number information, must be split into two separate **Street** and **HouseNumber** elements. Very often, this splitting is guided by ad-hoc rules, which make splits based on the element values.
- Transforming the attribute values.
- Restoring necessary syntactic formatting according to the target document standard.

The rules, which try to carry out the complete transformation process in one shot, have proven to be very complex. This causes serious problems in implementing and maintaining them. These problems are due to the mixture of several different aspects of the overall transformation process. For this reason, any re-use of such rules is practically impossible. Moreover, defining these rules from the scratch requires a great deal of manual effort. To overcome this bottleneck we have developed a multi-layered framework, which differentiates several aspects of the transformation process. Thus, we have been able to provide an approach where this translation is achieved by selecting and concatenating simple transformation rules. In short, we are able to transform a complex programming task into a simple plug-and-play process where the straightforward rule patterns are selected, instantiated, and combined. This multi-layered integration is discussed in the next section.

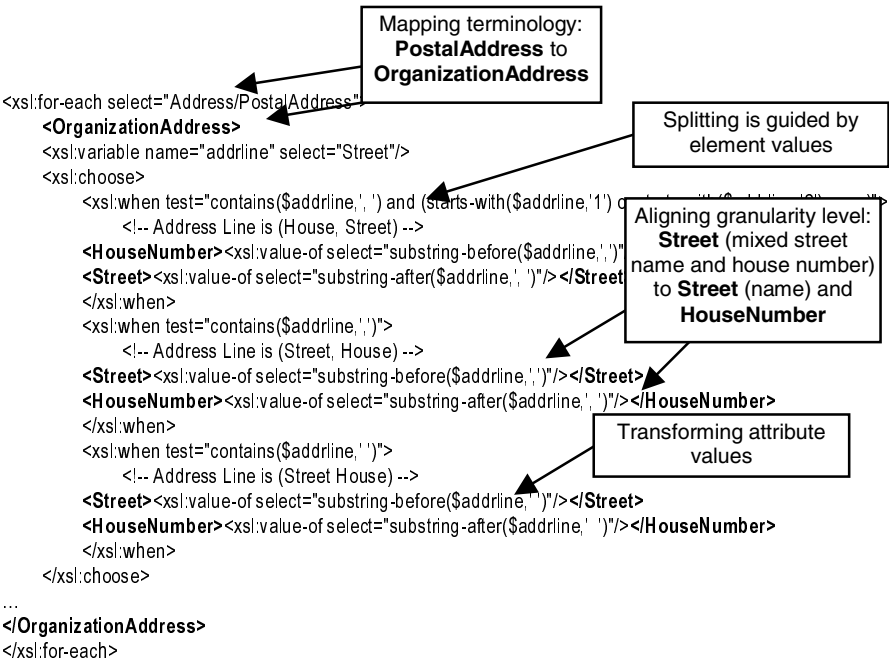


Fig. 3. Direct translation of the cXML address to the xCBL format. The elements, which form the resulting document, are presented in a bold font.

4 The Multi-layered Model for Catalog Integration

The problems of single-layer integration occur because two tasks run together with a single set of transformation rules: syntactical translations between different XML representations and semantic mapping between the terminology and granularity level of the representations. Naturally, these two types of transformations belong to different layers of representation.

The layered approach for information representation on the Web was proposed in [13], where three layers, a syntax layer, an object layer, and a semantic layer are proposed for information modeling on the Web. The syntax layer provides a way of serializing information content into a sequence of characters according to some standard, e.g. XML. The purpose of the object layer is to offer an object-oriented view of the information with the normalized data models of standardized triples. Finally, the semantic layer provides a conceptual model for the information. We have based our integration architecture on this partitioning.

Hence, we separate three layers for the catalog integration task as presented in Fig. 4. These include the *Syntax layer*, the *Data Models layer*, and the *Ontology layer*.

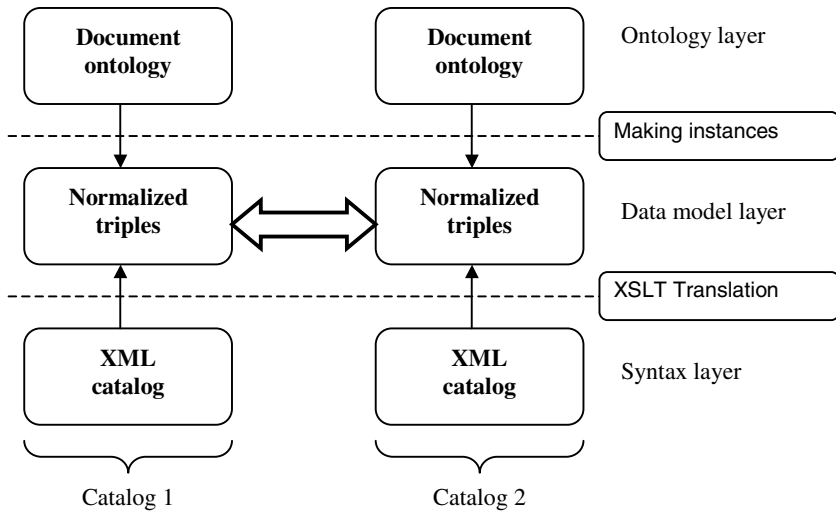


Fig. 4. The model for catalog information

The Syntax layer corresponds to the instance documents represented with their XML serialization. The serialization specifies used XML elements and attributes, and their order. Even semantically equal documents may differ in their serialization.

The Data models layer serves as a bridge between the Ontology layer and the Syntax layer. On this layer, the representations are abstracted from the differences imposed by the Syntax layer and the products are represented by object-property-value triples, where the properties stand for document elements.¹⁰ Normalization is done according to the corresponding ontology, which specifies, for example, that the Street cXML element actually represents the street and house number information and must be split accordingly.

The terminology used at this layer is defined by the corresponding ontology and generally has to coincide with the terminology used at the Syntax layer. However, the former might be more detailed than the latter, e.g. the XML serialization may allocate only one element for the street name and house number, while the ontology has to allocate two separate elements. The lowest detail level also binds the quality of the mapping of a pair of catalogs. This problem emerges when we need to incorporate a new catalog into a marketplace. We cannot map it without information lost if the new catalog provides more details than the ontologies already used in the marketplace. Hence, we should always assume that the ontology provides the most detailed partitioning possible. According to the ontology, the data model must maintain these

¹⁰ These object-property-value triples may also be used to represent instance-SubclassOf-class triples according to the ontology defined at the ontology layer.

details even if they are not currently needed, as they may be required later in mapping a new catalog.

We assume that different terminologies must be aligned at the Ontology layer rather than at the Data Models layer. We will discuss this more extensively in upcoming papers as a future enhancement of the two-layered approach discussed here.

In our approach, we used RDF [11] on the Data Models layer as the language to encode the triples. RDF is a W3C [1] standard for describing machine-processable semantics of data that is also represented by the object-property-value triples. However, RDF is not the only language available for this layer. Another suitable candidate is Simple Object Access Protocol (SOAP) [2] (see [8] for a comparison of SOAP with RDF).

The Ontology layer is presented at the top of Fig. 4. It defines a terminology used to represent the information provided by various product and document standards. This layer specifies the terminology in detail, sufficient to define the transformations between the catalogs with one-to-one mapping rules, as shown in the next section.

In addition, the ontology contains the elements specified as optional and possibly absent in the XML serialization and, therefore, helps in aligning them. Although we occasionally refer to this layer throughout the paper, we do not present any further discussion of the possible ontology mismatches or integration problems, which may arise in this layer (see [10] for a relevant discussion). In this paper, we refer to it only as a pre-requisite in defining atomic elements for describing the pieces of exchanged information.

5 Two-Layered Information Integration

As mentioned earlier, simultaneous execution of several integration tasks causes the difficulties of the single-layered integration model. For this reason, we have used a ‘divide-and-conquer’ approach to decompose these tasks into several sub-tasks, each of which is performed separately.

The decomposition is performed in a similar way to the structure of heuristic classification proposed in [2]. Heuristic classification assumes that the classification is performed on a layer of abstract structures, and the input data must be first abstracted, i.e. translated from some particular format into abstract solution classes; after the classification, this intermediate solution must be refined to specific solutions.

The integration involves two layers: the Syntax layer and the Data Models layer. RDF documents have a standard XML serialization, and we treat the representations from both layers as XML documents and use XSL-T rules to transform them. However, there are many standard RDF (XML) serializations for the same set of triples. Even more, RDF is much more than an XML serialization for triples (and even

¹¹ www.w3c.org

¹² www.w3.org/TR/SOAP/

not the best choice for a serialization). Enriched with RDF Schemas¹³ which specify the structure of RDF triples, RDF Data Models layer must be regarded as a *modeling* layer, rather than an intermediate layer. Hence, ‘pure’ RDF-based technologies must be used at the Data Models layer.

In both layers, information is presented in XML, because RDF documents from the Data Models layer are serialized in XML; hence it is natural to use XSL-T language to define the transformation rules.

Translation of a catalog requires three steps as depicted in Fig. 5 (1) mapping the catalogs into their data models during the *abstraction* step; (2) mapping between the data models during the *transformation* step; (3) translating the new model into the target XML format during the *refinement* step. These three steps are described in the following subsections.

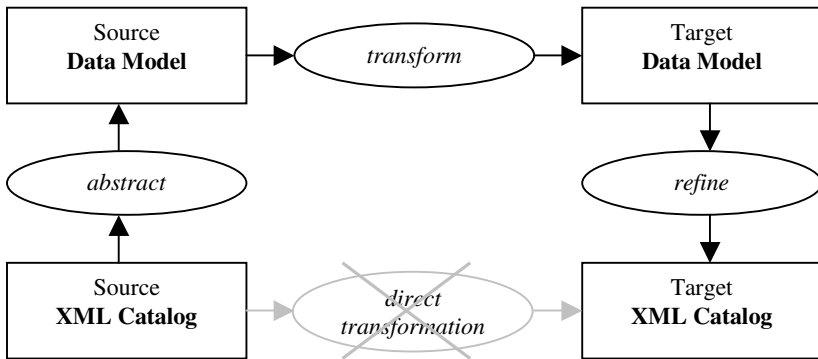


Fig. 5. The model for data transformation

Abstraction Step

At the abstraction step, the XML catalogs are translated into their data model encoded with RDF triples. This requires the following transformations:

- Translation of each XML element or XML attribute, which refers to a product feature into an RDF property with the same name.
- Split of a single XML element into two or more RDF triples, as specified by the corresponding ontology.
- Inclusion optional XML elements in the RDF triples, as specified by the ontology; the values of the elements are filled with the default value also specified in the ontology.
- Concatenation of multi-file descriptions into a single file.

At this stage, the transformations remain in the same or a more detailed terminology as specified by the input catalog. Hence, the abstraction step consists only of *one-to-many* (including *one-to-one*) mappings. The abstraction rules attempt to

¹³ www.w3.org/TR/2000/CR-rdf-schema-20000327/

align the catalog to a more detail representation, hence, no *many-to-one* rules may appear at this step.

One-to-one transformation is the simplest and most common type of transformation of an XML element into the corresponding RDF triple. Actually, it performs a syntactical translation between the XML and RDF serializations of the same element. For example, the following rule translates the cXML element `PostalAddress` and `PostalCode` to the xCBL elements `OrganizationAddress` and `PostalCode`:

```
<xsl:element name="rdf:Description">
  <xsl:for-each select="PostalAddress">
    <OrganizationAddress>
      ...
      <PostalCode><xsl:value-of select="PostalCode"/></PostalCode>
    </OrganizationAddress>
  </xsl:for-each>
</xsl:element>
```

One-to-many mapping occurs when the ontology specifies several elements, which are represented with a single element in the XML serialization. The XSL-T language provides the means to represent mapping XML elements and attributes, as well as possibilities to analyze text inside an element and to split it into two or more. XSL-T uses the XPath language to perform these operations. Accordingly, XSL-T rules must be extended with small XPath expressions (element parsers) which perform necessary element splits. For example, in the following fragment of an cXML address, it is assumed that the element `Street` contains street name separated from the following house number with a comma:

```
<Street>De Boelelaan, 1081a</Street>
```

We must split `Street` into two XML elements

```
<Street>De Boelelaan</Street>
<HouseNumber>1081a</HouseNumber>
```

as specified in the cXML ontology. This can be done with the following XSL-T rule:

```
<xsl:element name="rdf:Description">
  <OrganizationAddress>
    <xsl:variable name="addrline" select="Street"/>
    <Street><xsl:value-of select="substring-before($addrline, ',')"/></Street>
    <HouseNumber><xsl:value-of select="substring-after($addrline, ',')"/></HouseNumber>
  ...
</OrganizationAddress>
</xsl:element>
```

The above example, as well as the example from Fig. 3 presents only a few ways to place a street name and a house number in a single line. For example, an assumption that the street name starts from a letter used in Fig. 3 would fail to recognize the house number in the line ‘5-th Avenue 5’. Additional information might also be needed. In consequence, our approach sub-divides this step into two sub-steps: (1) transformation of the content presented by separate XML tags; here we can employ the semi-formal structure of the content; (2) wrapping natural language sequences and splitting them into separate information units; here we rely on the wrapper and information extraction technology applied to unstructured information sources.¹⁴

Transformation Step

All inter-catalog mappings are performed on the layer of RDF data models. We assume that the ontologies specify the catalogs on a detail level sufficient to specify all inter-catalog mappings with *one-to-one* transformations, where all necessary element splits are performed during the abstraction step (Fig. 5) and the necessary element merges will be done on the refinement step. Hence, only *one-to-one* mappings may appear on this step.

One-to-one mappings are done in the same way as at the abstraction step. At the transformation step, the mappings actually perform syntactical transformations between the RDF serializations of the data models of the catalogs. The transformation rules may appear as follows (from cXML to xCBL):

```
<xsl:when test="@about='cXML'">
  <xsl:attribute name="about">xCBL</xsl:attribute>
    <OrganizationAddress><xsl:value-of select="PostalAddress"/></OrganizationAddress>
    ...
</xsl:when>
```

Refinement Step

During the refinement step, all syntactical restrictions required by the target format are restored, and necessary many-to-one transformations are performed:

- Each RDF triple is serialized with a corresponding XML element or attribute.
- One or more RDF triples have to be merged into a single XML element, if required.
- Target XML elements are created in proper order.
- The target XML representation may be partitioned into several files, if required.

In consequence, only *many-to-one* (including *one-to-one*) rules may appear at this step. Both types of rules can be easily implemented with XSL-T without XPath fragments.

¹⁴ See *RISE: Repository of Online Information Sources Used in Information Extraction Tasks*: <http://www.isi.edu/~muslea/RISE/> for a survey on the available technology in this area.

One-to-one mappings are done in a straightforward way with XSL-T rules which transfer the RDF triples into the XML catalog elements and attributes. In the example shown in Fig. 6 the PostalCode element is translated into XML with the one-to-one mapping.

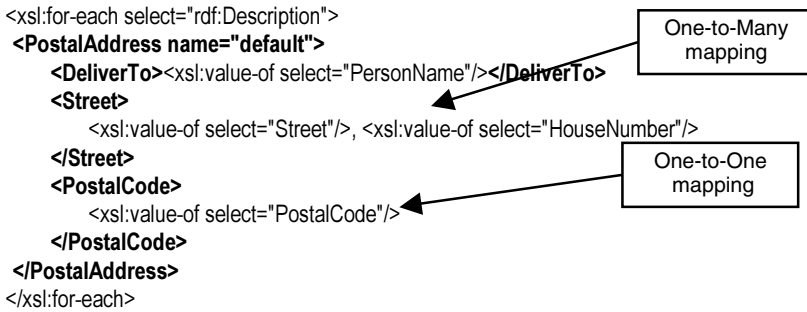


Fig. 6. Refinement rules for cXML

Many-to-one mapping occurs when two or more RDF attributes must be translated to one XML element or attribute. For example, the cXML ontology stores the street name and the house number separately for simplicity of inter-catalog mappings discussed earlier. However, the XML serialization assumes that both elements must be split into a single `Street` element. This is done by means of XSL-T, merging the elements

```

<Street>De Boelelaan</Street>
<HouseNumber>1081a</HouseNumber>

```

into the following cXML element:

```

<Street>De Boelelaan, 1081a</Street>

```

as shown in Fig. 6. Street creation rule.

6 Conclusions

In this paper, we sketched the problems, which occur in the catalog integration task when approached directly without intermediate sub-steps relaying on a multi-layered approach (see [15] for more details). We presented a two-layer integration approach (implicitly referring to a third layer, which we will discuss in the upcoming papers). The introduction of the second layer makes it possible to decompose the integration task into three simpler tasks: *abstraction*, *transformation*, and *refinement*. Each of the tasks can be resolved and debugged separately. Up to a certain degree, this approach overcomes the main problem of the information integration task in B2B electronic commerce. The problem lies in the high complexity and unreadability of the rules, which still have to be developed manually. In our approach, such translation can be

achieved by selecting, instantiating, and combining elementary translation rule patterns.

At present, we are working on a framework and tool environment, which allows effective and efficient definition of such mappings. This framework must provide:

- A simple language on top of XSLT, which is customized to the specific needs of mapping rules in electronic commerce. Instead of manually defining transformation directly in XSLT, they should be derivable from the mappings defined at a more intuitive level. This helps to transform a complex programming task into a simple plug-and-play process based on the simplified rule patterns, identified by separating different mapping aspects.
- We used XSLT rules to translate the data models throughout the paper. The data models layer is created to hold ontology instances. They are encoded with RDF triples and must conform an ontology (schema), most likely represented with RDF Schema. Future development of the architecture requires dealing with RDF and RDF Schema querying, verification of RDF to RDF Schema, and performing inference on the Schemas. We will examine possible ways to exploit the Sesame¹⁵ tool (cf. [1] for a state of the art report) for this process.
- Finally, the integration architecture will consist of several ontologies, each of which will specify a certain aspect of B2B mediation. They are: content standards, content aligning ontologies, document ontologies (which instances from the data model discussed in this paper), partner codes (e.g., UDDI¹⁶), and a workflow ontology specifying the way how all other ontologies must interact together to make a B2B marketplace functioning.

Successful B2B electronic commerce must deal with three serious mapping problems: (1) different content standards define (in different ways) over 10,000 classes and ten times more attributes for describing products; (2) different document standards define (in different ways) over 400 business documents exchanged in electronic trading; (3) different product catalog standards define (also in different ways) complex structures for describing the products exchanged. In short, B2B marketplaces face a need for the development of an appropriate technology, which allows them to easily define their large number of complicated mappings. Otherwise, they will suffer the same fate that left the tower of Babel an unfinished ruin.

Acknowledgements. We would like to thank Hans Akkermans, Guy Botquin, Ying Ding, Stefan Decker, Michel Klein, Ellen Schulten, Volodymyr Zykov, and three anonymous referees for their insightful comments, which helped us in writing this paper.

References

- [1] Broekstra, J., Fluit, C., van Harmelen, F.: The State of the Art on Representation and Query Languages for Semistructured Data. IST-1999-10132 On-To-Knowledge Project, Deliverable 8, August (2000); available online at: <http://www.ontoknowledge.org/del.shtml>

¹⁵ <http://sesame.aidministrator.nl/>

¹⁶ www.uddi.org

- [2] Clancey, W.: Heuristic Classification, *Artificial Intelligence* **27** (1985) 289-351
- [3] Clark, J.: XSL Transformations (XSL-T), W3C Recommendation, November (1999); available online at <http://www.w3.org/TR/xslt/>
- [4] Clark, J., DeRose, S.: XML Path Language (XPath), version 1.0, W3C Recommendation, November 16 (1999); available online at <http://www.w3.org/TR/xpath>
- [5] Commerce One, Inc.: Commerce One XML Common Business Library (xCBL) 3.0, Press release made at the eLink Conference, Hong Kong, November 29 (2000); available online at <http://www.commerceone.com/news/us/xcbl30.html>
- [6] Fensel, D.: *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*, Springer-Verlag, Berlin (2001)
- [7] Fensel, D., van Harmelen, F., Akkermans, H., Klein, M., Broekstra, J., Fluit, C., van der Meer, J., Schnurr, H.-P., Studer, R., Davies, J., Hughes, J., Krohn, U., Engels, R., Bremdahl, B., Ygge, F., Reimer, U., Horrocks, I.: OnToKnowledge: Ontology-based Tools for Knowledge Management, In: *Proceedings of the eBusiness and eWork 2000 Conference (EMMSEC-2000)*, Madrid, Spain, October 18-20 (2000)
- [8] Haustein, S.: Semantic Web Languages: RDF vs. SOAP Serialization, In: *Proceedings of the Workshop on the Semantic Web - SemWeb2001 at the 10-th WWW Conference*, Hong Kong, May 1 (2001)
- [9] Integrated generic resource: *Fundamentals of product description and support*, International Standard ISO 10303-41, Second Edition (2000)
- [10] Klein, M.: Combining and relating ontologies: an analysis of problems and solutions. In: Gomez-Perez, A., Gruninger, M., Stuckenschmidt, H., Uschold, M. (eds.): *Proceedings of the Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001)*, Seattle, USA, August (2001); available online at <http://www.cs.vu.nl/~mcaklein/>
- [11] Lassila, O., Swick, R.: *Resource Description Frame-work (RDF) Model and Syntax Specification*, W3C Recommendation, February (1999); available online at <http://www.w3.org/TR/REC-rdf-syntax/>
- [12] Li, H.: XML and Industrial Standards for Electronic Commerce, *Knowledge and Information Systems* **2** (2000) 487-497
- [13] Melnik, S., Decker, S.: A Layered Approach to Information Modeling and Interoperability on the Web, In: *Proceedings of the Workshop on the Semantic Web at the Fourth European Conference on Research and Advanced Technology for Digital Libraries (ECDL-2000)*, Lisbon, Portugal, September 21 (2000)
- [14] Open Applications Group Inc.: *Open Applications Group Integration Specification*, OAGIS Release 7.0.2 (2000); available online at <http://www.openapplications.org/>
- [15] Omelayenko, B., Fensel, D.: An Analysis of the Integration Problems of XML-Based Catalogues for B2B Electronic Commerce, In: *Proceedings of 9th IFIP 2.6 Working Conference on Database Semantics*, Hong Kong, April 25-28 (2001); available online at <http://www.cs.vu.nl/~borys/papers/>
- [16] Omelayenko, B., Fensel, D.: An Analysis of B2B Catalogue Integration problems: Content and Document Integration, In: *Proceedings of the International Conference on Enterprise Information Systems (ICEIS-2001)*, Setúbal, Portugal, July (2001); available online at <http://www.cs.vu.nl/~borys/papers/>
- [17] U.S. Department of Commerce: *Digital Economy 2000*, White paper, June (2000)

A Visual One-Page Catalog Interface for Analytical Product Selection

Juhnyoung Lee¹, Priscilla Wang², and Ho Soo Lee¹

¹ IBM T. J. Watson Research Center
P. O. Box 218, Yorktown Heights, NY 10598, U.S.A.
{jyl, leehs}@us.ibm.com

² Department of Computer Science, University of California at Berkeley
Berkeley, CA 94720, U.S.A.
pwang@cory.eecs.berkeley.edu

Abstract. One of the key elements of e-commerce systems is the online product catalog. In this paper, we discuss the interface of online product catalogs, focusing on its ability to help shoppers navigate and analyze product information. Specifically, we present a new interactive interface for online product catalogs that is effective in navigating through the product information space and analytically selecting suitable products. It uses a multi-dimensional visualization mechanism based on parallel coordinates and augmented by a number of visual facilities for filtering, color-coding, and dynamic querying. We present a prototype and explain how the prototype visualizes the entire product information space in a single page and supports intuitive exploratory analysis with the visual facilities.

1 Introduction

Over the past few years, *electronic commerce* (or *e-commerce*) on the Internet has emerged as a dramatic new mode of business. One of the key elements of e-commerce systems is the *online product catalog*. Online product catalogs are much more than the electronic version of printed product catalogs. On one hand, they are a *content management system* that provides sellers with the ability to assemble, aggregate, and normalize product (or service) information from supplier databases and to quickly, inexpensively, and easily distribute the information. On the other hand, they provide potential buyers with an *interactive interface* by offering a multimedia representation of product information as well as retrieval, classification, and ordering services.

There have been extensive research and development activities for both aspects of the online product catalog. For content management, a number of commercial products are available from companies such as CardoNet, Interwoven, OnDisplay, Poet Software, Versifi, and Vignette, and are used by many e-commerce sites [5]. For catalog interfaces, various methods supporting product search and navigation have been developed and are being utilized. They include hierarchical browsing, search based on keywords or parameters, interactive product design or configuration, dialog-based product filtering, product ranking, and the side-by-side product comparison

table [6]. These product information navigation methods are sometimes referred to as *shopping metaphors*. The usability of several product selection mechanisms was empirically studied in [2, 6]. Also, the impact of different shopping metaphors on the effectiveness of online stores in terms of click-through and conversion rates was analyzed and visualized in our previous work [4].

In this paper, we revisit the interface aspect of the online product catalog focusing on its ability to help shoppers solve the *product selection problem*. When selecting a product (or service) to buy, a shopper often needs to consider a number of different factors. For example, there may be factors related to the *product specification* such as price, material quality and properties, color and size. In addition, there may be factors related to the *service specification* such as delivery time and cost, and warranty. Furthermore, there may be *supplier qualification* factors such as delivery experience and reputation. To make reasonable decisions about which products to buy, the shopper should be able to analyze and compare the multiple attribute values of alternative products (and their suppliers). In this sense, product selection is a *multi-criteria decision analysis* process [1]. One of the important functions of the online product catalog interface is to facilitate the analysis process by helping buyers effectively explore and analyze product information.

In this paper, we present a new interactive interface for online product catalogs for e-commerce sites such as online retail stores and marketplaces. The interface is referred to as *Visual One-Page Catalog* (VOPC). It uses a visualization mechanism based on *parallel coordinates* and augmented by a number of visual operations. In several ways, VOPC facilitates navigation of the product space and analysis of individual products taking their multiple attribute values into account. First, it presents the entire information space of offered products in a single Web page. This compact display makes it easy to explore the product space and compare different products. Despite the amount of product information shown in a single page, the visual presentation of VOPC alleviates the possibility of information cluttering. Diverse visual operations together with the visualization solve the navigation and analysis aspects of the product selection problem. Easy-to-use visual facilities for filtering, tagging, color-coding, and dynamic querying are effective for exploring the product information space.

The rest of this paper is structured as follows: Section 2 presents user scenarios of the status quo of online product catalogs. Section 3 describes the design and facilities of the VOPC interface. Section 4 revisits the user scenarios and explains how the VOPC interface would help with its capabilities for navigation and analysis. Finally, in Section 5, conclusions are drawn.

2 User Scenarios of Online Product Catalogs

The status quo of online product catalogs is the key motivator in the design of the Visual One-Page Catalog (VOPC). Typical online product catalogs are comprised of a number of product pages, Web pages that provide information about one or more products being sold. Each page usually only provides information about a small num-

ber of products, e.g. one to ten products, which tend to belong to the same or a similar product category. As a result, shoppers normally need to browse multiple product pages in order to accumulate information about one or more products of interest. This task in itself is a hassle. In addition, add the task of comparing different attributes (e.g., price, brand, style, etc.) of different products (e.g., shirt, pants, shoes, etc.), and the whole process of comparison-shopping on the Internet becomes cumbersome. Perhaps the best way to understand the problems with the majority of online product catalogs is by taking a close look at specific examples. Note that the scenarios are based on real interfaces at the time of research in June 2000.

2.1 An Online Retailer: Amazon.com

Amazon.com sells everything from 'A' to 'Z' and has the "Earth's Biggest Selection." The main page of Amazon's online product catalog is divided into five sections: a left column, a right column, folder tabs on the top, miscellaneous links on the bottom, and the main space that is left in the middle. The folder tabs at the top of the page correspond to the table of contents in a paper catalog, and provide links to the major sections of the catalog. In the left column, if a shopper knows roughly what s/he wants to buy, s/he can simply do a search on the item's product category or the item name itself. If the shopper only knows the general category s/he is interested in, s/he can browse the appropriate category. There are also links to shopping services (e.g., buying or redeeming gift certificates) and special features (e.g., purchase circles). In the right column, there is a section devoted to new releases, and a section listing items that have dramatically increased or decreased in recent sales. The main space in the center of the page is for ads, which usually have a particular theme (e.g., Teen Scene, Summer Stuff). Each sub-page features more ads and often includes a "Top Sellers" list. In general, an abundance of information is presented on each page, and the shopper must actively engage in shopping or risk being overwhelmed. Most of these design aspects are not specific to Amazon's online product catalog but are common to the majority of online product catalogs for online retailers.

User Scenario

Jason, a college student, wants to buy a cookbook that is designed for students who have little money, little time, and little cooking experience. He goes to Amazon.com and proceeds to browse the Books category, then Cooking, Food & Wine, then Quick & Easy. At this point, he can choose between three categories: General, Cooking for One, and Microwave Cookery. He decides to browse Cooking for One. There are 80 books in this category, which are spread out over four pages (25 books on each of the first three pages, and the last five books on the last page). Now Jason is faced with the task of deciding which book(s) to buy. He finds that it is relatively simple to compare two books that are described one after the other, but difficult to compare book descriptions that are far apart (both on the same page and on different pages). Jason must either make mental comparisons or jot down information. If he is interested in a number of the books, mental comparison is very difficult if not impossible. At the same

time, jotting down information for many books is time-consuming and a hassle. Alternatively, if Jason is clever, he could add the items he is interested in to the shopping cart, making it slightly easier to compare the books since only key information is included in tabular fashion in the shopping cart view (but then he must still go back to the original listing to see additional information). However, Jason would still have to rely on just numbers and text to make comparisons (i.e. there is no visual representation of the data to assist his decision-making process). Furthermore, once he has made up his mind, he must remove (delete) all items he does not want from the shopping cart or start over with a new (empty) shopping cart. After jotting down a list of 14 books (title, price, and average shopper review for each), Jason develops a cramp in his hand and although he still has two more pages of items to look at, he decides to just buy one of the items he has already written down. His mood at the ending of his shopping experience could be described as somewhat frustrated.

2.2 An Electronic Marketplace: VerticalNet

One of VerticalNet's features is its targeted communities, where professionals and companies in a particular industry can go for fast, efficient business information, interaction, and transactions. The focus here is on the manner in which transactions can be made between professionals and industries. Currently, VerticalNet sponsors 56 industry communities, which are grouped by industry type, and are accessible through links on VerticalNet's homepage. All of the community pages have consistent designs, making it relatively easy for a shopper to navigate through any community's page once the shopper has gotten the feel of one particular community. In addition, professional shoppers can search for auctions or fixed price deals through the link to *Industry Deals*.

User Scenario

A clerk at a dentist's office needs to order more x-ray film. The clerk, Melissa, decides to try ordering using one of VerticalNet's health care communities. Since this is her first time using VerticalNet, she must first search the product catalogs of each of the five health care communities. She finds a match for 'x-ray film' in only one of the five communities, *Hospital Network.com*. The match indicates that there are seven companies associated with *Hospital Network.com* that sell x-ray film. To automatically contact the listed companies to request a price quote, literature, a sample, customer service or service assistance, she must first register to be a member of the community. Alternatively, she can click on a link for a particular company to see a more comprehensive product list as well as the company's basic contact information or she can go directly to the company's storefront (a showcase for the company's business and products) if it has one. Ultimately, Melissa must register with the *Hospital Network.com* community to get the most out of VerticalNet. When registering, she selects 'e-mail' as her preferred contact method and 'immediate' as the urgency of the requested info. Within a day, she receives e-mails from six of the seven companies. She decides to call the seventh company the next day since she needs to place an order

as soon as possible and wants to make her order based on information from all possible companies. Finally, after collecting the seven different sets of information, she discovers that the companies sell x-ray film in two different units. She decides to make a table listing the type of film and the price per standard unit of film for each company. After placing an order with the company that offers the best deal, Melissa is satisfied that she found the best buy. However, she feels that the whole buying process was relatively lengthy, and wonders if she should have just placed an order with the company that the office bought x-ray film from in the past, even though that company's rate for x-ray film is slightly higher than this other company's rate.

The two case scenarios should have manifested the motivation driving VOPC. It would not be too surprising if Jason and Melissa decide to go to a brick-and-mortar store the next time they need to purchase something. The VOPC interface offers an alternative structure to the typical online product catalog with the goal of making online shopping experiences more pleasant for shoppers like Jason and Melissa. The structure of VOPC involves compacting many product pages into one page using parallel coordinates. The use of parallel coordinates not only saves shoppers from needing to navigate endless product pages, but also assists shoppers in the multi-attribute decision-making process involved in choosing which product(s) to buy.

3 The VOPC Interface

A parallel coordinate system was proposed as a more practical way of displaying multi-dimensional data sets [3]. Visualization of higher dimensional geometry with the traditional Cartesian coordinate system where all axes are mutually perpendicular is difficult. In parallel coordinates, all axes are parallel to each other and equally spaced. In an n -dimensional space, there are n parallel axes (X_1, X_2, \dots, X_n) each representing a separate variable. To plot an n -dimensional point (x_1, x_2, \dots, x_n), adjacent values on their respective vertical axes are connected by a line for a total of $(n-1)$ connected line segments. In this way, parallel coordinates allow the visualization of higher order geometries in an easily understood two-dimensional representation. VOPC uses parallel coordinates to display products and their attributes in a single Web page. Each axis represents an attribute; the online shoppers dynamically determine the number and type of attributes. The plot of a (multi-dimensional) point represents a product. The VOPC prototype has been developed in the Java programming language using IBM VisualAge for Java IDE (Interactive Development Environment) and IBM DB2 Universal Database System. The prototype runs as a Java applet because VOPC is to be used in an Internet context. For implementing graphical elements of the VOPC interface, we used the JFC Swing classes.

3.1 Product Categories

The product categories represent the different types of products in a catalog, or how products are organized in the product catalog. For example, two categories in an

automobile catalog are vehicle type and manufacturer. In the VOPC interface, each category is associated with a hierarchical view of the products. Shoppers can view the different tree structures by clicking on their preferred category of organization in the tabbed pane on the left side of the user interface. Each category is also associated with a different color-coding of the products. If all categories requested do not have any sub-categories, then the products are color-coded according to their category. However, if any category has sub-categories, then products are color-coded by one category only.

3.2 Product Attributes

The product attributes are features that a shopper considers when purchasing a product. As mentioned earlier, they include features related to product specification, service specification, and supplier qualification. In this design, we group those features into three tiers based their importance. Attribute values in the first tier are crucial in the decision analysis process and/or have known most desirable values (i.e., most desirable values do not vary from shopper to shopper). For example, in an automobile catalog, first-tier attributes would include price and ratings. Attributes in the first tier are included in the parallel coordinates plot as axes. Second-tier attributes are important, but do not fit in the first tier (i.e., lower or higher boundary values are not necessarily more desirable). In an automobile catalog, second-tier attributes might include an image of a car and fuel tank capacity. Attributes in the second tier are either included in the parallel coordinates plot with the use of target values, or displayed if the user decides to view additional information about a product. Attribute values in the third tier are relatively unimportant to the typical shopper. In an automobile catalog, third-tier attributes might include number of cylinders and braking distance. Third-tier attributes are displayed if the user decides to view additional information about a product.

3.3 Catalog Operations

VOPC provides a user interface that a shopper can use to request the initial product catalog view. Using this interface, the shopper can set bounds on attribute values to narrow the scope of products s/he is interested in possibly buying. For example, the shopper may wish to look at only those cars that have MSRP values between \$24,000 and \$40,000. In this case, the shopper must simply enter those two values as the lower and upper, respectively, MSRP bounds. Then when the associated product view is displayed, the MSRP axis is bounded by those two values. If the shopper does not specify bounds for an attribute in the initial catalog view request, the minimum and maximum values for that attribute for all products in the catalog are used for the bounds. Figure 1 displays a product catalog view in the VOPC model showing five first-tier attribute values of ten products from three different product categories (distinguished by the color used for the “matchsticks”) and two product lines. Initially, the product lines are not shown in the VOPC view; product lines can be added to the view

by using “tagging” operations as will be explained later. The names of the attributes and the unit of their values are specified below each axis. The attribute values displayed as “matchsticks” in the view are color-coded by product category. Also, the visualization of attribute values helps the shopper intuitively understand how the product attribute values are distributed over individual attribute coordinates.

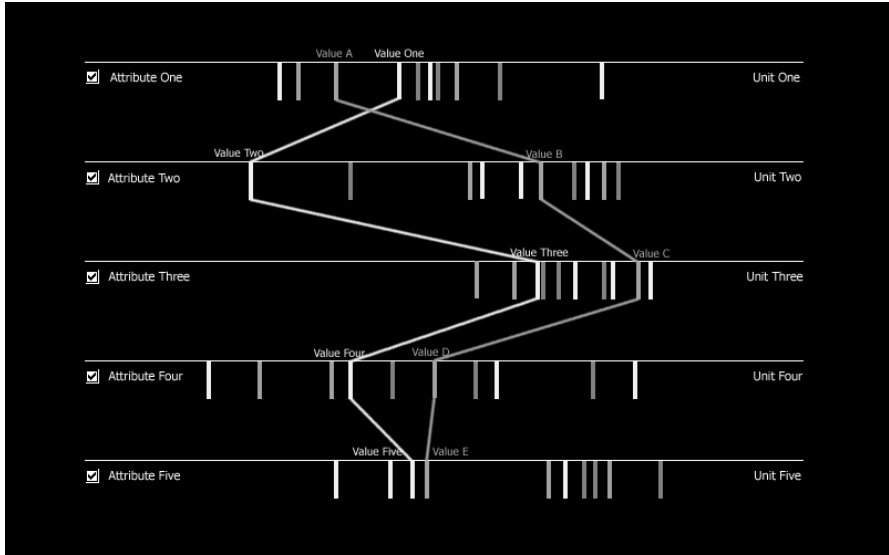


Fig. 1. A product catalog view

VOPC allows the shopper to dynamically choose all the attributes to show in the initial product catalog view. Initially, all checkboxes (each associated with an attribute) are selected. However, if the shopper later decides that a particular attribute is not that important in the analytical product selection process (e.g., if all products share the same value for an attribute), s/he can de-select the checkbox for that attribute. De-selected attributes are moved to the end of the list of attributes, and are not part of product lines. The corresponding axis and points are moved accordingly. Selecting a de-selected checkbox moves the checkbox and its corresponding axis and points to the end of the list of attributes that are already selected. The shopper can do a series of de/select operations on the attribute checkboxes to order the attributes by decreasing importance. Doing so would aid the shopper’s analysis process. The operation of changing the order of the attributes can be simplified by adding drag and drop capabilities to the checkboxes.

A product line is the line formed as a result of plotting a point on the parallel coordinates axes. Each product has an associated product line. Only a subset of product lines is visible at a time. The subset is determined by where the mouse pointer is positioned on the VOPC screen. If the mouse pointer were positioned over a data point (which is represented by a “matchstick”), all product lines that share that data point would be made visible. When a product line triggered by a mouse fly-over is displayed

in the VOPC view, a tool-tip showing the short description of the product can also be shown.

One of the primary goals of the VOPC interface is to help the shopper analyze product information by comparing alternative products. Therefore, the layout of the product lines is key to the design of the VOPC interface. For the shopper to fully take advantage of VOPC, s/he must be aware of this layout. On each attribute axis, the values either increase or decrease according to whether higher or lower values, respectively, are more desirable in general. In this fashion, the product lines visually show their product's desirability. That is, product lines that are closer to the right side of the screen are more desirable than product lines that are on the left side of the screen. The right side of the screen was chosen to be the "target area" because the left side is already somewhat cluttered with attribute checkboxes.

It is likely that no product line will appear completely in the target area on the right side of the screen. In this case, ordering attributes by importance helps the shopper to assess which products are most desirable. By placing the most important attributes towards the top of the screen, the shopper knows that a product line that fits in the target area in the top part of the screen and wanders out of the target area in the bottom part of the screen is more desirable than a product line that fits in the target area in the bottom but not top part of the screen. The shopper can use this kind of placement/desirability information to make a good decision about which product(s) to buy. Figure 1 shows the comparison of two products from different categories in the VOPC model.

Since a product line is only displayed when the mouse pointer is positioned over one of the product line's data points, the tagging operation relieves the shopper of having to rely on visual memory when comparing two or more product lines. To tag a product line, right-click on any data point that is part of the product line and select "Tag." A pop-up menu window is displayed upon a right-click on a data point for tagging and other operations ("More Info" and "Add to Cart") associated with individual product lines. If multiple product lines share a data point and that data point is used as the starting point of the tag operation, the shopper is prompted to choose which product line(s) to tag. Tagged product lines remain visible in the catalog view even when the mouse pointer is no longer positioned over one of the product line's data points. Product lines may also be untagged. To un-tag a product line, right-click on any data point that is part of the product line and de-select "Tag."

4 User Scenarios: Revisited

In this section, we revisit the user scenarios presented in Section 2. We explain how the VOPC interface would help Jason and Melissa in the buying process.

4.1 How VOPC Would Help Jason at Amazon.com

The Visual One-Page Catalog could have saved Jason a considerable amount of time and effort and provided him with a better overall shopping experience at Ama-

zon.com. In Jason's case, the only category of VOPC would be *Cooking for One* (with a total of 80 products). (Had Jason been unable to choose between the three sub-categories of the *Quick & Easy* category, all three sub-categories would be included as categories in the visual product catalog view. If Jason later decided that he did not want to browse books in one or two of these categories, he could simply filter them out.) As for the attributes, Jason would be able to choose the factors that he considers most important to his decision. For example, the main attributes might be title, price, and average shopper review (number of stars). In addition, Jason could customize his search by specifying attribute ranges. For example, he might only be interested in cookbooks that are under \$15 and have received an average shopper review of at least four out of five stars. Jason would also be able to specify whether a low attribute value or a high attribute value is more desirable, and the product catalog would then order the products on each attribute line accordingly. In this case, a low price and a high average shopper review are most desirable, so values on the price attribute line would, from left to right, decrease from high to low respectively, and values on the average shopper review attribute line would increase from low to high. As a result, when Jason is comparing product lines, his goal is to find a product line that is as far right as possible, as that would represent the best buy for him. To stress the relative unimportance of the title, Jason could de-select the title attribute. Doing so would cause any visible product lines to not extend to the title axis. If Jason weighs the importance of each main attribute, VOPC could even rank, say, the three best cookbooks for him. By having Jason himself choose which attributes are important, this prevents the catalog from initially displaying any unnecessary information. Jason can easily view additional information about products he is particularly interested in. In this case, additional information might include author, number of pages, shipping time, availability, percent savings, a picture of the book's cover, publication date, and reviewer's comments. By saving Jason time and effort, VOPC would very likely encourage Jason to return to the Web site to make additional purchases in the future.

4.2 How VOPC Would Help Melissa at VerticalNet

There are a number of ways the Visual One-Page Catalog could improve Melissa's online shopping experience. Ideally, VerticalNet would need to gather information for products by category within each industry group (for all communities and companies). At the very least, for the VOPC model to be useful in this scenario, each community should gather information for products by category from all the companies associated with the particular community. With the VOPC model, Melissa would simply search for 'x-ray film', which would result in the generation of a catalog that contains different kinds of x-ray film (whose companies may or may not belong to different communities within the health care group). The categories of VOPC would be different kinds of x-ray film. The main attribute of VOPC would be price per standard unit. Another attribute might be shipping time. A zooming in operation on a data point could bring up the name of the company that offers that particular product. Clicking on a data point could bring up an order form for that product. This basic functionality of the VOPC model would save Melissa time in two key ways. First, she would not have to

search for 'x-ray film' multiple times (first in the five different health care communities, then in the seven companies associated with *Hospital Network.com*). Second, the visualization method provided by the VOPC interface would save Melissa from needing to make a table comparing the different products since the relationship between the different products could be understood simply by looking at the shapes and positions of the different product lines. (In this case, the product lines may actually be single points since the only attribute Melissa may be interested in is price.)

5 Concluding Remarks

In this paper, we have presented a new interactive interface for online product catalogs that provides an alternative structure to the typical online product catalog. It uses a multi-dimensional visualization mechanism based on parallel coordinates and augmented by a number of visual facilities for filtering, tagging, color-coding, and dynamic querying that are useful for exploring and analyzing product information. The structure of VOPC involves compacting product pages into a single Web page. In just a single page, the shopper can visually compare all the products s/he is interested in by looking at the different product lines of the VOPC interface. VOPC helps shoppers by visually showing them concrete comparison data that might otherwise be only abstractly in their heads or on paper in non-visual form. Only essential information in the decision-making process is represented in the product lines; however, more detailed information is accessible through operations with the mouse pointer on the data points of any product line. The shopper determines what is essential information and what is extra information. In this way, the VOPC interface functions as an online shopper's assistant in the decision-making process and results in the shopper having a more pleasant online shopping experience.

References

1. J. C. Anderson and J. A. Narus, *Business Market Management: Understanding, Creating and Delivering Value*, 1st Edition, Prentice Hall, Inc., November 1998.
2. E. Callahan and J. Koenemann, "A Comparative Usability Evaluation of User Interfaces for Online Product Catalog", *Proceedings of the 2nd ACM Conference on Electronic Commerce*, 2000.
3. A. Inselberg and B. Dimsdale, "Parallel Coordinates A Tool for Visualizing Multivariate Relations", *Human-Machine Interactive Systems*, Plenum Publishing Corporation, 1991.
4. J. Lee, M. Podlaseck, E. Schonberg, and R. Hoch, "Visualization and Analysis of Clickstream Data of Online Stores for Understanding Web Merchandising", *International Journal of Data Mining and Knowledge Discovery*, 5(1), January 2001, Kluwer Academic Publishers.
5. A. Neumann, "A Better Mousetrap Catalog", *Business 2.0*, February 2000, pp. 117-118.
6. M. Stolze, "Comparative Study of Analytical Product Selection Support Mechanisms", *Proceedings of INTERACT 99*, Edinburgh, UK, 1999.

Engineering High Performance Database-Driven E-commerce Web Sites through Dynamic Content Caching

Wen-Syan Li, K. Selçuk Candan, Wang-Pin Hsiung,
Oliver Po, and Divyakant Agrawal

CCRL, NEC USA, Inc., 110 Rio Robles M/S SJ100, San Jose, CA 95134, USA
{wen,candan,whsiung,agrawal}@ccrl.sj.nec.com

Abstract. The fast growing demand for e-commerce brings a unique set of challenges to build a high performance e-commerce Web site both in technical terms and in business terms. To ensure the fast delivery of fresh dynamic content and engineer highly scalable e-commerce Web sites for special events or peak times continuously put heavy pressures on IT staffs due to complexity of current e-commerce applications. In this paper, we analyze issues related to engineering high performance database-driven e-commerce web sites including: (1) integration of caches, Web servers, application servers, and DBMS; and (2) tradeoff of deploying dynamic content caching versus not deploying. We describe available technology in the scope of *CachePortal* project at NEC. We illustrate performance gains through our technology using an e-commerce Web site built based on some of the most popular components, such as Oracle DBMS, BEA WebLogic Application Server, and Apache Web server.

1 Introduction

Forrester Research Inc. [1] expects that by 2002 over 47 million people will purchase goods and services online in the United States alone and the U.S. Internet commerce will grow to \$327 billion by that same year. The fast growing demand for e-commerce brings a unique set of challenges to build high performance e-commerce Web sites both in technical terms and in business terms.

In technical terms, to ensure the fast delivery of fresh dynamic content and engineer highly scalable e-commerce Web sites for special events or peak times continuously put heavy pressures on IT staffs due to complexity of current e-commerce applications. In business terms, the brand name of an e-commerce site is correlated to the type of experience users receive. The need for accounting for users' quality perception in designing Web servers for e-commerce systems has been highlighted by [2]. A typical database-driven Web site consists of the following components:

1. A database management system (DBMS) to store, maintain, and retrieve all necessary data and information to model a business.

2. An application server (AS) that incorporates all the necessary rules and business logic to interpret the data and information stored in the database. AS receives user requests for HTML pages and depending upon the nature of a request may need to access the DBMS to generate the dynamic components of the HTML page.
3. A Web server (WS) which receives user requests and delivers the dynamically generated Web pages.

One possible solution to scale up database-driven e-commerce sites is to deploy network-wide caches so that a large fraction of requests can be served remotely rather than all of them being served from the origin Web site. This solution has two advantages: serving users via a nearby cache closer to the users and reducing the traffic to the Web sites. Many content delivery network (CDN) vendors [3,4] provide Web acceleration services. The study in [5] shows that CDN indeed has significant performance impact. However, for many e-commerce applications, HTML pages are created dynamically based on the current state of a business, such as product prices and inventory, rather than static information. As a result, the time to live (TTL) for these dynamic pages can not be estimated in advance. As a result, content delivery by most CDNs are limited to handling fairly static pages and streaming media rather than the full spectrum of dynamic content discussed in this paper.

To coordinate cache servers, WS, AS, and DBMS and ensure fast delivery of fresh dynamic contents is challenging since these servers and DBMS are independent components. And, currently there is no effective and efficient mechanism to ensure that database content changes are reflected to the caches. As a result, most e-commerce sites have to specify dynamic contents as non-cacheable. Consequently, each request to an e-commerce site results in both network delay and server delays (i.e. WS delay, AS delay, and DBMS delay) since the request must be processed each time at the Web site.

In [6], we propose a framework for enabling dynamic caching. Here we further address the following issues: (1) integration of cache servers, WS, AS, and DBMS; and (2) tradeoff of deploying dynamic content caching versus not deploying. We also illustrate performance gains of dynamic content delivery using our approach through a e-commerce Web site built based on some of the most popular e-commerce Web site components, such as Oracle DBMS [7], BEA WebLogic Application Server [8], and Apache Web server.

The rest of this paper is organized as follows. In Section 2, we give an overview of the system architecture of a typical database-driven e-commerce Web site and identify factors that impact response times. In Section 3, we present a loosely-coupled approach that we have developed for e-commerce Web site acceleration. In Section 4, we provide cost analysis on our technology and discuss the tradeoff between deploying caches and not deploying. In Section 5, we report experimental results. Finally we discuss related work and give our concluding remarks.

2 Architecture and Scalability Issues of E-commerce Web Sites

In this section, we start by presenting the typical architecture of e-commerce Web sites and then identify the bottlenecks that limit the scalability of such architectures. The architecture of a typical e-commerce Web site consists of has back-end systems, such as DBMS or file systems, that store and maintain the most up-to-date information to model the business and the current state of the business. The application server as described earlier incorporates the necessary rules to capture the business logic and retrieves necessary data by querying the DBMS or the file-system. This data is then processed as per the business logic to dynamically generate the HTML pages. In addition to the requests from the Web and the application server, the DBMS at a database-driven Web site also receives updates (through the Web or back-end processes) to the underlying data in the databases.

When a user accesses the Web site, the request and its associated parameters, such as the product name and model number, and cookie information for customization, are passed to an application server. The application server performs necessary computation to identify what kind of data it needs from the database or file system, or external data sources. Then the application server sends appropriate queries to the database or other sources. After the database returns the query results to the application server, the application uses these to prepare a Web page and passes it to the Web server, which then sends it to the user.

Caching has been viewed as an effective way to scale up Web sites. The percentage of the dynamically generated Web pages on the Internet is getting higher. As a matter of fact, dynamic contents count for more than 25 percent of overall traffic within NEC networks. Unfortunately, since the application servers, databases, Web servers, and caches are independent components, today, there is no efficient mechanism to make database content changes to be reflected to the cached Web pages. Since, most e-commerce applications are sensitive to the freshness of the information provided to the clients, most application servers have to specify dynamically generated Web pages as *non-cacheable* or make them expire immediately. Consequently, repeated requests to dynamically generated Web pages with the same content result in repeated computation in the back-end systems (application and database servers).

Apache Web server can serve static content requests up to more than 2000 requests per second and BEA WebLogic Application Server can also serve static content requests up to 1000 requests per second. However, for the Web site serving dynamic content using BEA WebLogic and Oracle 8i, the performance is less scalable: it can serve traffic up to 300 requests per second in our testing. This is understandable since Web servers and application servers are light-weight programs designed for specific purposes. On the other hand, database management systems are "*heavy weight*" general purpose software. Thus, based on the experimental results presented here, we believe that the appropriate approach to Web acceleration for database-driven e-commerce applications should be to *reduce the load on the database* or to *reduce the frequency of DBMS accesses*.

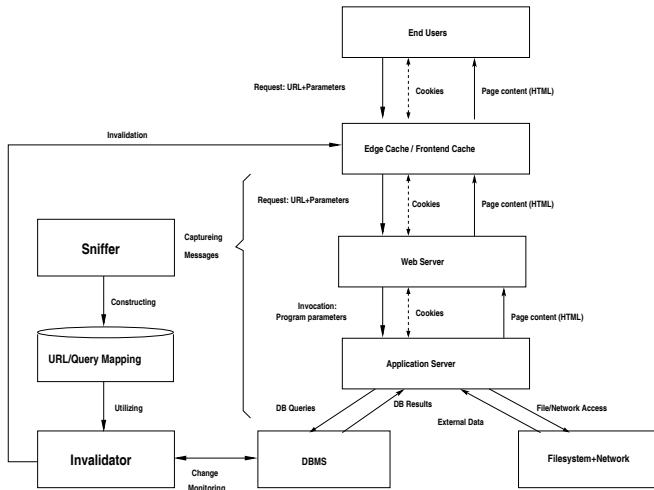


Fig. 1. Architecture of a Database-Driven E-Commerce Site with *CachePortal*

3 *CachePortal* - Technology for Integrating Caches, Web Servers, Application Servers, and DBMS

In this section we describe an open architecture for accelerating delivery of e-commerce content over wide-area networks. The architecture and underlying technology, referred to as *CachePortal*, enables caching of dynamic data over wide-area networks.

3.1 System Architecture

In Figure 1, we show an architecture and data flows in a database-driven e-commerce Web site which employs the **CachePortal** technology. In Figure 1, we illustrate the communication between the caches, the Web server, the application server, and the database as follows: The Web server communicates with the application server using URL strings and cookie information, which is used for customization. The application server communicates with the database using queries. The database changes can be monitored by the tuples and query statements recorded in the database log. However, the database itself, which knows how and what data changes, does not know how the data is used to generate dynamic Web pages. In addition, the database does not know which dynamic content, identified by URLs, is impacted by these database changes.

Our proposed technology *enables* dynamic content caching by deriving the relationships between cached pages and database access via a *sniffer*, and intelligently monitoring database changes to "eject" related pages from caches via an *invalidator*. Note that knowledge about dynamic content is distributed across three different servers: the Web server, the application server, and the database management server. Consequently, it is not straightforward to create a mapping

between the data and the corresponding Web pages automatically in contrast to the other approaches [9,10,11,12], which assume such mappings are provided by system designers.

The sniffer creates the URL to database query mapping (shown in Figure 1). Sniffer collects the query instances, but it does not interpret them. The URL information is gathered either before the Web server using a module which listens to the incoming HTTP requests or before the application server using a module which uses the environment variables set by the application server. The Query/URL map contains (1) a unique ID for each row in the table representing the Query/URL map (2) the text of the SQL query to be processed by the invalidator and the list of tables in the query, and (3) the URL information, including the HTTP_HOST string, a list of (cookie,value) pairs, a list of (get variable name,value) pairs, and a list of (post variable name,value) pairs.

The invalidator, on the other hand, listens to the updates in the database and using the Query/URL map, identifies pages to be invalidated and notifies the relevant caches about the *staleness* of the cached page. For this purpose, it interprets the query instances in the QI/URL map. The invalidator consists of three subcomponents. The first component periodically examines the database log to extract the updates since the previous extraction. The second component analyzes the updates (which are in terms of inserts, deletes, and modifications of tuples to specific tables) and identifies the queries that are impacted. These queries are registered in the Query/URL map. The third component identifies the URLs that are invalidated due to impacted queries and generates cache invalidation messages which are destined to the network-wide caches via the API provided by the CDNs. We now show the invalidation process using two examples.

Example 1. Let us assume that we have an e-commerce application which uses a database that contains two tables, *Car*(maker, model, price) and *Mileage*(model, EPA). Let us also assume that the following query, *Query1*, has been issued to produce a Web page, say *URL1*:

```
select maker, model, price from Car where maker = "TOYOTA";
```

If we observe that a new tuple (*Toyota*, *AVALON*, 25,000) is inserted into (or deleted from) the table *Car* in the database, we would know that the results of *Query1* is impacted, and consequently *URL1* needs to be invalidated.

Example 2. In the above example, since the query contains only one table, it is possible to evaluate the impact without any additional information. On the other hand, if we assume the following query *Query2* which has been issued to produce *URL2*,

```
select Car.maker, Car.model, Car.price, Mileage.EPA
from Car, Mileage
where Car.maker = "TOYOTA" and
      Car.model = Mileage.model;
```

Since *Query2* needs to access two tables, we can check whether a new tuple inserted into *Car* does not satisfy the condition in *Query2* without any additional information. But if the new tuple satisfies the condition, then we cannot check whether or not the result is impacted until we check the rest of the condition, which includes the table *Mileage*. That is, we need to check whether or not that the condition *Car.model* = *Mileage.model* can be satisfied. To check this condition, we can issue the following polling query, *PollQuery*, to the database:

```
select Mileage.model, Mileage.EPA from Mileage
where "AVALON" = Mileage.model;
```

If there is a result for *PollQuery*, we know that the new insert operation has an impact on the query result of *Query2* and consequently *URL2* needs to be invalidated.

4 Tradeoff between Deploying CachePortal Dynamic Content Caching and Not Deploying

Ensuring freshness of dynamic content in caches does put additional workload to databases due to monitoring database changes and invalidation. We now examine the extra load that polling queries puts on the database management system in greater detail. In particular, we would like to estimate whether the overhead of polling queries on the DBMS completely offsets the savings we realize due to the caching of dynamic content in the network components.

Assuming that the DBMS workload we obtain is that the access rate, A . And the update rate is $U \times A$ and read only access rate is $(1 - U) \times A$. Let the cache hit ratio be H . Then, $(1 - U) \times A \times H$ accesses to the DBMS are served from the caches, which is the *benefit* of deploying **CachePortal**.

Now let us examine the *cost* introduced by the invalidator to execute polling queries for multi-relation queries. We now model the characteristics of the workload parameters at the invalidator. The polling query workload will depend upon the number of queries in the Query/URL map, which represents the URLs that have been cached in the network.

One major factor that generate the polling queries is the fraction of queries that involve multiple relations although queries that involve single relation do not require execution of polling queries. The cost introduced by the invalidator is $U \times A \times \text{Cost_Factor}$. The more polling queries can be issued and executed for each update, the higher is the *Cost_Factor*. Let us assume that each update would result in 50 polling queries which can be executed in time within two accesses to the DBMS can be executed. To execute a query, almost 98% of time is connection time. For executing polling queries, we can use persistent connection to execute 50 polling queries at cost equivalent to two units of connection time in our system setting. In this case, *Cost_Factor* is 2. Thus, the net benefit of deploying *CachePortal* is $(1 - U) \times A \times H - U \times A \times \text{Cost_Factor}$ access workload is removed from the DBMS.

There is a tradeoff between using caches and not using caches. Since synchronization would put load to databases, deploying caches would not necessarily be

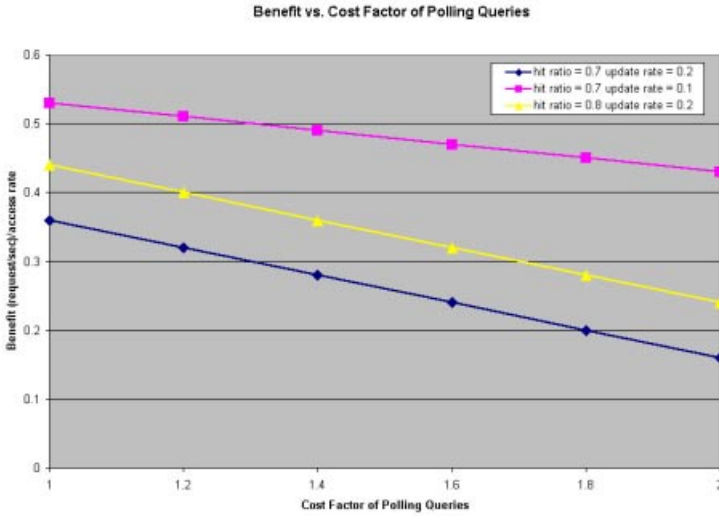


Fig. 2. Benefit of Deploy *CachePortal* with the Respect to Polling Query Cost Factor

beneficial if content update rates are too high or cache hit rates are too low. In Figure 2, we illustrate the benefit of deploy *CachePortal* technology with respect to polling query cost factor. As we can see, as the cost factor for polling queries increases, the benefit (percentage of requests that are served from caches) reduces. A lower update rate or a higher hit ratio will increase the benefit.

5 Experiments on Evaluating Effectiveness of *CachePortal* Technology

We now report the results of our experimental evaluation of the *CachePortal* technology under different settings. We considered the following two architectural configurations:

Case 1: the Web server and the application server are located on the same machine. and database is located on the separate machine. No cache is used.

Case 2: the Web server and the application server are located on the same machine. The database is located on the separate machine. *CachePortal* technology is applied and a front-end cache, a cache close to servers, is used. The freshness of cached content is guaranteed.

The database used in this experiment contains 7 tables and 25,000 records. The update rate is 1 per second and each dynamic content page request results in a query with one join operation to the database. All machines are Pentium III 700 Mhz CPUs running Redhat Linux 6.2 with 1 GByte of main memory. Oracle

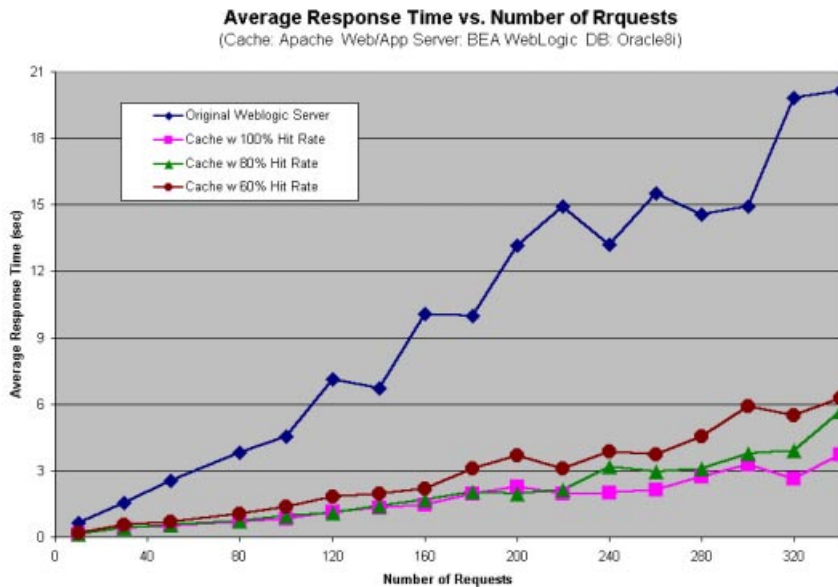


Fig. 3. Evaluation on User Response Time

8i is the DBMS employed. BEA WebLogic Application Server is used as Web servers and application servers. Apache Web server is modified to be used as a cache server.

We recorded both the user response time (i.e. round trip time observed by the users) and the server latency (i.e. the time difference between the time when the requests arrive at the Web server and the time when the requests are served from the Web server). Our results indicate that the response time in case 1 increases sharply when the number of HTTP requests increases. On the other hand, the response time in case 2 remains relatively flat for the settings where the hit ratios are 60%, 80%, and 100% respectively. By analyzing response times and server latencies in each case we found that the user response time delay is mainly caused by the server latency rather than the network delay. This means that the requests do arrive at the server without much network delay, but they are queued at the application server waiting for query results from the database.

6 Related Work

Caching of dynamic data has received significant attention recently [13,14]. Dynamai [9] from Persistence Software is one of the first dynamic caching solution that is available as a product. However, Dynamai relies on proprietary software for both database and application server components. Thus it cannot be easily incorporated in existing e-commerce framework. Challenger et al. [10,11,12] at

IBM Research have developed a scalable and highly available system for serving dynamic data over the Web. In fact, the IBM system was used at Olympics 2000 to post sport event results on the Web in timely manner. This system utilizes database triggers for generating update events as well as intimately relies on the semantics of the application to map database update events to appropriate Web pages. Our goal in this paper is to design similar update capability but in the context of general e-commerce setting. Compared with the work [10,11,12], where the relationships between Web pages and underlying data are specified manually, our approach automatically generates the query/URL mapping.

[15,16] propose a diffusion-based caching protocol that achieves load-balancing; [17] uses meta-information in the cache-hierarchy to improve the hit ratio of the caches; [18] evaluates the performance of traditional cache hierarchies and provides design principles for scalable cache systems; and [19] which highlights the fact that static client-to-server assignment may not perform well compared to dynamic server assignment or selection.

SPREAD[20], a system for automated content distribution is an architecture which uses a hybrid of *client validation*, *server invalidation*, and *replication* to maintain consistency across servers. Note that the work in [20] focuses on static content and describes techniques to synchronize static content, which gets updated periodically, across Web servers. Therefore, in a sense, the invalidation and validation messages travels horizontally across Web servers. Other works which study the effects of *invalidation* on caching performance are [21,22,23]. Consequently, there has been various cache consistency protocol proposals which rely heavily on *invalidation* [20,24,25]. In our work, however, we concentrate on the updates of data in databases, which are by design not visible to the Web servers. Therefore, we introduce a *vertical* invalidation concept, where invalidation messages travel from database servers and Web servers to the caches.

7 Concluding Remarks

In this paper, we analyze the issues related to engineering high performance database-driven e-commerce web sites including: (1) integration of caches, WS, AS, and DBMS; (2) tradeoff of deploying dynamic content caching versus not deploying; and (3) how to design a "cache friendly" Web site. We believe that with CachePortal technology, a well-designed cache friendly Web site can be engineered as a high performance e-commerce Web site where the freshness of the Web pages delivered are ensured.

References

- [1] Forrester Research Inc., <http://www.forrester.com/>
- [2] N. Bhatti, A. Bouch, and A. Kuchinsky. *Integrating User-Perceived Quality into Web Server Design*, International World Wide Web Conference, WWW9, Amsterdam, The Netherlands, pp. 1-16, 2000.
- [3] Akamai Technology, <http://www.akamai.com/html/sv/code.html>
- [4] Digital Island, Ltd., <http://www.digitalisland.com/>

- [5] B. Krishnamurthy and C.E. Wills, *Analyzing Factors That Influence End-to-End Web Performance*, International World Wide Web Conference, WWW9, Amsterdam, The Netherlands, pp. 17-32, 2000.
- [6] K. Selçuk Candan, Wen-Syan Li, Qiong Luo, Wang-Pin Hsiung, and Divyakant Agrawal. Enabling Dynamic Content Caching for Database-Driven Web Sites. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, Santa Barbara, CA, USA*, ACM Press, 2001.
- [7] Oracle Corporation, <http://www.oracle.com/>
- [8] BEA Systems Inc., <http://www.bea.com/>
- [9] Persistence Software, <http://www.persistence.com/dynamai/>
- [10] Jim Challenger, Paul Dantzig, and Arun Iyengar. *A Scalable and Highly Available System for Serving Dynamic Data at Frequently Accessed Web Sites*, in Proceedings of ACM/IEEE Supercomputing'98, Orlando, Florida, November 1998.
- [11] Jim Challenger, Arun Iyengar, and Paul Dantzig. *A Scalable System for Consistently Caching Dynamic Web Data*, in Proceedings of IEEE INFOCOM'99, New York, New York, March 1999.
- [12] Eric Levy, Arun Iyengar, June-hwa Song, and Daniel Dias. *Design and Performance of a Web Server Accelerator*, in Proceedings of IEEE INFOCOM'99, New York, New York, March 1999.
- [13] Fred Douglass, Antonio Haro, and Michael Rabinovich. *HPP: HTML Macro-Preprocessing to Support Dynamic Document Caching*, In Proceedings of USENIX Symposium on Internet Technologies and Systems, 1997
- [14] Ben Smith, Anurag Acharya, Tao Yang, Huican Zhu. *Exploiting Result Equivalence in Caching Dynamic Web Content*, In Proceedings of USENIX Symposium on Internet Technologies and Systems, 1999
- [15] A. Heddaya and S. Mirdad. *WebWave: Globally Load Balanced Fully Distributed Caching of Hot Published Documents*, ICDCS 1997.
- [16] A. Heddaya, S. Mirdad, and D. Yates *Diffusion-based Caching: WebWave*, NLNR Web Caching Workshop, 9-10 June 97.
- [17] M.R. Korupolu and M. Dahlin. *Coordinated Placement and Replacement for Large-Scale Distributed Caches*, IEEE Workshop on Internet Applications, pp. 62-71, 1999.
- [18] R. Tewari, M. Dahlin, H.M. Vin, and J.S. Kay. *Beyond Hierarchies: Design Considerations for Distributed Caching on the Internet*, ICDCS 99.
- [19] R.L. Carter and M.E. Crovella. *On the Network Impact of Dynamic Server Selection*, in Computer Networks, 31 (23-24), pp. 2529-2558, 1999.
- [20] P. Rodriguez and S. Sibal, *SPREAD: Scaleable Platform for Reliable and Efficient Automated Distribution*, International World Wide Web Conference, WWW9, Amsterdam, The Netherlands, pp. 33-49, 2000.
- [21] D. Wessels, *Intelligent Caching for World-Wide Web Objects*, Proceedings of INET-95, 1995.
- [22] P. Cao and C. Liu, *Maintaining Strong Cache Consistency in the World Wide Web*, IEEE Transactions on Computers, 47(4):445-457, Apr. 1998.
- [23] J. Gwertzman and M. Seltzer. *World-Wide Web Cache Consistency* In Proceedings of 1996 USENIX Technical Conference, pages 141-151, San Diego, CA, January 1996.
- [24] H. Yu, L. Breslau, and S. Shenker. *A Scalable Web Cache Consistency Architecture*, In Proceedings of the ACM SIGCOMM'99, Boston, MA, September 1999.
- [25] D. Li and P. Cao. *WCIP: Web Cache Invalidation Protocol*, 5th International Web Caching and Content Delivery Workshop, Poster Session, Lisbon, Portugal, 22-24 May 2000.

XML Enabled Metamodeling and Tools for Cooperative Information Systems

Christophe Nicolle and Kokou Yétongnon

Equipe Ingénierie Informatique, Laboratoire Le2i
Faculté des Sciences Mirande, Université de Bourgogne
BP 47870, 21078 Dijon cedex, FRANCE
{cnicolle, kokou}@u-bourgogne.fr

Abstract. The development of tools to support semantic resolution is a key issue in the design of heterogeneous interoperable information systems. This paper presents a methodology and a data model translator toolkit, called X-TIME, for the design and management of interoperable information systems. X-TIME combines a metamodeling approach and XML features to provide support for automated design of wrappers or semantic reconciliators. It is a flexible semantics oriented approach that takes into account several important characteristics of interoperable information systems, including extensibility and composability. Extensibility requires a translation scheme that can easily integrate new data models. Composability, which is the ability of an application to dynamically determine the data source it needs, requires on demand translation of data models. To meet these requirements, X-TIME is based on an extensible metamodel that can be used to represent meta-level semantic descriptors of existing modeling concepts.

1 Introduction

Traditionally, database design methodologies have been directed towards the development of techniques and tools for precise representation and efficient storage of data in centralized information systems. With the emergence of wide area networks and web oriented technologies, more and more enterprises and organizations that are based on distributed and disconnected data sets will need integrated information systems. Therefore, new design solutions are required to meet the challenges created of these emerging systems. The objective is to support the amalgamation of autonomous information systems into coherent interoperable information systems to share data and services and carry out processing tasks cooperatively. Consequently, there has been a shift of research and design focus towards developing methodologies and techniques for accessing local as well as remote information sources.

Several approaches have been proposed for the design of these integrated collections of information systems, ranging from distributed systems to federated systems, language based multibase systems and mediation based federations. A distributed system can be viewed as a global system in which data is accessed

via a global schema. Schema integration techniques are required to create the global schema [4]. A federated system consists of a set of heterogeneous databases in which federation users access and manipulate data transparently without a knowledge of their location [12]. A language multi-base system consists of loosely connected databases in which a common query language is used to access the contents of the participating databases [8]. In this approach, in contrast to distributed and federated systems, users have to know data location. Mediation systems require wrapper and mediator components to resolve syntactic and semantic differences between the components information sources of interoperable systems [5]. This approach is more adapted to web based environment where extensibility and reusability are important. A major issue that must be addressed in most approaches involves the development of concepts and tools to bridge semantic gaps among component systems. Most solutions are to some extent based on schema integration, data model mapping or semantic reconciliation techniques. Data model mapping can typically be used to translate the original database schema to a common representation before others techniques are applied [11].

In this paper, we present a methodology and tools framework, called X-TIME, to support the design and management of interoperable information systems. It is a data model translator toolkit, based on a metamodel and XML technology, that is aimed at facilitating the design of wrappers or semantic reconciliators. X-TIME is an adaptable semantics oriented meta-model approach that takes into account important characteristics of interoperable information systems, including extensibility and composability. Extensibility requires a translation scheme that can easily integrate new data models while composability, which is the ability of an application to define dynamically the subset of data sources it needs, requires on demand translation among a subset of data models. To meet these requirements, X-TIME is based on an extensible metamodel that defines a set of metatypes for representing meta-level semantic descriptors of existing modeling concepts. The metatypes are organized in a generalization hierarchy to capture semantic similarities among modeling concepts and correlate constituent data models of interoperable systems. X-TIME evolves from a design methodology called TIME that was initially developed in 1996 [10]. An overview of metatype hierarchy is presented in section 2. A description of our methodology is given in section 3. Finally section 4 concludes the paper.

2 Scenario of Cooperation

To illustrate the presentation of the X-TIME methodology, consider the collection of heterogeneous information systems. It includes 3 sites corresponding to two hotels information systems and a tourism office or travel agency decision support system. The tourism office's site is a web-based hotel reservation system running on a relational DBMS (Oracle 7.3.2). The hotel information systems run on a relational DBMS (Progress 7.3) an object oriented (O2) and DBMS respectively. The cooperating information systems must meet the following re-

quirements. Every site retains control and autonomy over its data and services. The hotels use their information systems to make room reservations locally. Finally, the tourism office wishes to define an open decision system capable of accommodating and integrating new hotel information systems regardless of the format of their DBMS. As stated above, the main issue that must be addressed is to support the resolution of semantic discrepancies among component systems. Many solutions require an intermediate model to map local schema to a global common representation [2,3,7]. In this work, we use a specific metamodel based on a generalization hierarchy to support semantic reconciliation of heterogeneous data models. Figure 1 depicts the main components of a metatype hierarchy for defining translators and semantic reconciliation components to support a cooperation between object oriented and relation databases. The generalization hierarchy includes several metatypes: a top-level generic metatype called META, object-oriented metatypes that are meta-descriptors for concepts that are used to model objects with complex or simple structure and inter-object association links, and specific metatypes that are used to model relational, and object data model concepts. Some metatypes are derived from other metatypes by specialization. A metatype M is defined by a tuple $M = (A_M, C_M, P_M)$, where A_M model syntactic elements used to describe the structures of real world entities. C_M represents a set of constraints on the structural descriptors of A_M , and P_M is a (possibly empty) set of operations associated with M . The extensible meta-model contains two groups of metatypes. The kernel of the metamodel is made up of five basic generic metatypes: META, MComplex-Object (MCO), MSimple-Object (MSO), MNaryLink (MNL) and MBinary Link (MBL). The second group contains a set of data model specific metatypes. They are used to extend the metamodel when new data models are introduced in the multidatabase system. In our example, MClass (MCLA) and Minheritance-Link (MIHL) model object data model concepts and MRelation (MREL) models relational data model concepts. A detailed description of the metamodel and its main features is presented in [3].

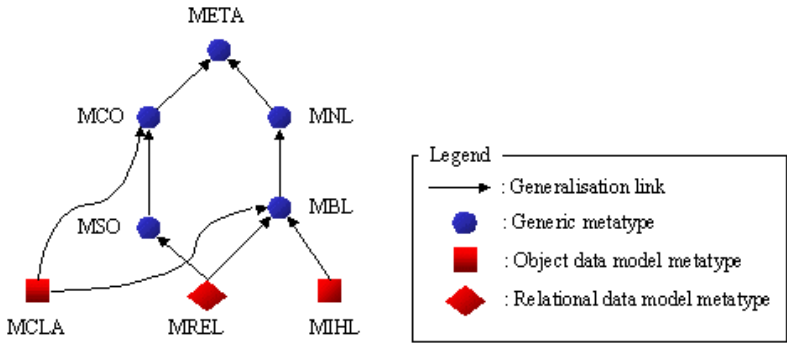


Fig. 1. Specific metamodel for Relational and Object data models

To specify the characteristics and basic semantics of the metatypes composing the metamodel, we use the XML-DATA language [11,15]. This language allows the description of information using tags without syntactic description. At a meta-level, the syntactic description of a concept or a schema is not represented. This description is carried out in a different stage by a "syntax association" process that tailors a generic XML based metatype description to the specific characteristics of a data model. This is done from the meta-level to the local-level using XML style-sheets [13]. Figure 2 shows the general components and representation levels of the XML based translation and semantic reconciliation process. It depicts two levels of representation. The cooperation level is devoted to XML based documents that describe the semantic properties of metatypes while the local level contains style-sheet based descriptions. Each style sheet description document contains data model specific descriptions that are used to complete the generic XML based metatype descriptions of the cooperation level. The use of style-sheets greatly reduces the heterogeneity problems. For example, in figure 2 the translation of a schema from the Progress information system to another relational RDBMS requires the same meta-schema but two different style-sheets to take into account differences between the syntax of Progress and Oracle databases. Note that style-sheets can be used to resolve differences between different releases of the same DBMS.

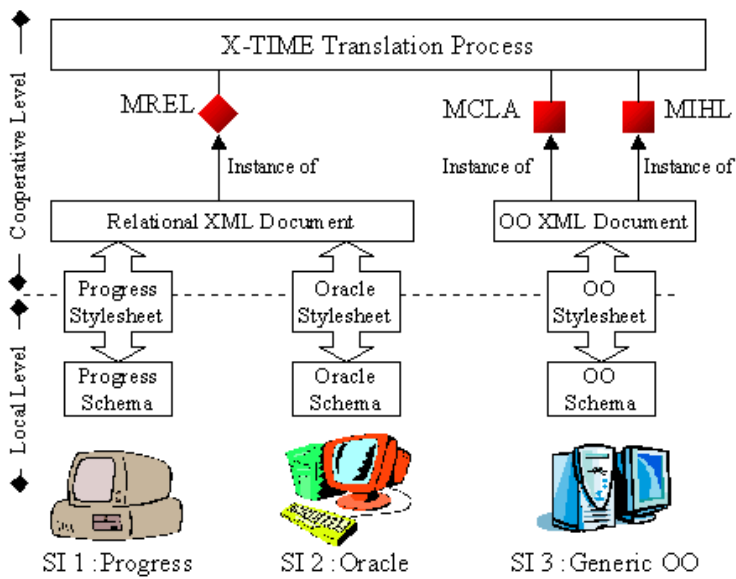


Fig. 2. Different level of representation

3 Overview of the X-TIME Methodology and Support Tools

This section presents an overview of the translation steps and the tools that are used to support the design of cooperative information systems. Three main tools are provided to support the development of interoperable architectures. An XML based editor (X-Editor) that can be used by a site manager to describe data models. A Java based Translation Rule Builder (TRB) that provides a set of functions to support metatype manipulation rules. Finally a Translator Compiler that permits the automatic construction of translators. A detailed description of these tools is beyond the scope of this paper. The descriptions are given in [9].

Three main steps are required to use the X-Time method to support a co-operation of heterogeneous information systems. The first step concerns the description of data models of participants of the cooperation and the construction of the metamodel hierarchy to establish semantic relations among data models of the component information systems. The second step is a rewriting step in which transformation rules are defined between set of metatypes. The last step is devoted to the generation of translators needed to reconcile semantics to exchange data.

3.1 Step 1: Description of Data Model Concepts

To participate in a cooperation, the data models of the component information systems must be described in an XML data format. A data model description tool, called X-Editor, is defined for this task. It allows the description of local data model concept at both semantic and syntactic levels. The semantic description of the data model concepts is made using pre-defined XML-Data concepts defined in [14]. The syntactic description of the data model concepts is made using the XML style-sheet Language XSL [13]. A piece of XML Style-sheet code is presented in figure 3. This style-sheet creates an XML document for a relational script. The script uses relational directives such as "CREATE TABLE", "NOT NULL", etc.

The descriptions created by X-Editor are based on metatypes defined in the generalization hierarchy of the X-TIME metamodel. As stated above, the kernel of the metamodel comprises the following metatype descriptions.

- **Object Metatypes:** Two basic metatypes are included in the kernel of the metamodel to represent object oriented modeling constructs, which can be used to describe real world entities. Metatype MComplex-Object characterizes concepts used to model complex-structure real world data while metatype MSimple-Object categorizes simple or flat structure modeling construct such as the table concept of the relational model. The set of attribute descriptors associated with Object metatypes allow the representation of concepts used to model 1) primitive types (integer, sting, char) and 2) complex types defined by complex structure constructors (set, list,...).

- **Link Metatypes:** To model links which abstract different relationships or associations among real world entities, the kernel contains two basic metatypes.

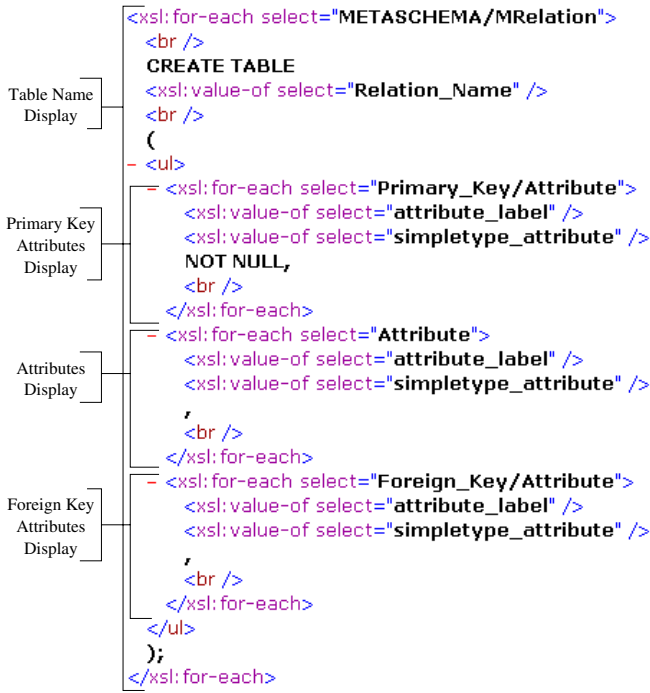


Fig. 3. XML Style-sheet for Relational script

Metatype MNary-Link represents the category of modeling types for n-ary relationships. Occurrences of this metatype are the relationship types of the ER model and the specialization / generalization links found in E.R and OO models. Metatype MBinary-Link represents modeling concepts, which are used to express binary relationships between two real world entities. In addition, link metatypes can also model 1) cardinalities associated with objects participating in relationship and 2) specific attributes of relationship types.

• **Specific Metatypes:** The metatype MRelation (REL) represents the concept RELATION of the Relational data model. It is defined by $REL = (A_{REL}, C_{REL}, P_{REL})$. MRelation specializes the metatype MSimple-Object. Thus, the structure A_{REL} is identical to the structure of A_{SO} . MRelation refines the ID function defined in META to correspond to the precise definition of the relational primary key; namely, a sub-sequence of attribute labels that uniquely identify the tuples of a relation. Moreover, since a relation may in fact represents a relationship between two or more real world entities, MRelation also specializes the metatype MBinary-Link. Therefore, an edge is added between metatypes MRelation and MNarylink, and the component C_{REL} of MRelation must include constraints inherited from the links metatypes. The corresponding XML-DATA definition is presented in figure 4.

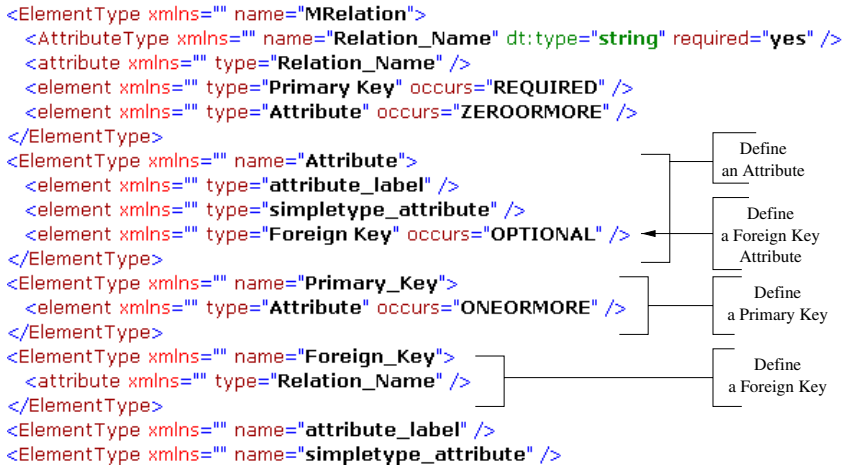


Fig. 4. XML-Data definition of the MRelation metatype

Meta-types MClasse (MCLA) and Minheritance-Link (MIHL) represent the object-oriented concepts. As illustrated in figure 1 the meta-type MClass specializes meta-type MComplex-Object and MBinary-Link. It is defined by $MCLA = (A_{MCLA}, C_{MCLA}, P_{MCLA})$. The structure A_{MCLA} is a mixed between the structure of A_{MMCO} and the structure of A_{MBL} . The inheritance from A_{MBL} allows the representation of reference attributes. Conversely, The meta-type MClasse refines the component P_{MCO} to introduce the behavioral aspect of the objects. A detailed and formal description of methods is beyond the scope of this paper. Thus the component P_{MCLA} is not presented. C_{MCLA} is empty. The corresponding XML-DATA definition is presented in figure 5.

The meta-type Minheritance-Link is a binary link. In addition to the constraints inherited from meta-type MBinary-Link, it maintains a subset constraints between the population of the specialized and generalized meta-types which generalizes the concept of inheritance of the object oriented data model. Moreover, Minheritance-Link specializes the structure of MBinary-Link to represent connection without attribute. The meta-type Minheritance-Link is defined by $MIHL = (A_{MIHL}, C_{MIHL}, [])$.

3.2 Step 2: Transformation Rules

The hierarchy of specialization defined in the previous step can be used to reduce the number of transformation rules needed to build the translators. Rather than defining rules between all metatypes, it is sufficient to define rules between pairs of directly connected metatypes (figure 1).

Transformation rules convert the XML-DATA schema of a source metatype to the XML-DATA schema of a target metatype. To build transformation rules, we have developed a Java based tool called Translation Rule Builder (TRB). This tool provides a set of functions to support the definition of rules that

```

<ElementType xmlns="" name="MClass">
  <AttributeType xmlns="" name="Class_Name" dt:type="string" required="yes" />
  <attribute xmlns="" type="MClass_Name" />
  <element xmlns="" type="Cla_Attribute" occurs="ZEROORMORE" />
</ElementType>
<ElementType xmlns="" name="Cla_Attribute">
  <element xmlns="" type="attribute_label" />
- <group groupOrder="OR">
  <element xmlns="" type="simpletype_attribute" occurs="ONEORMORE" />
  <element xmlns="" type="complextypes_attribute" occurs="ONEORMORE" />
  <element xmlns="" type="Class_Name" occurs="ONEORMORE" />
</group>
</ElementType>

```

Fig. 5. XML-Data definition of the MClass metatype

manipulate metatypes. These functions are simple functions such as change, replace, delete, create, insert, and find a string in a file, and complex functions such as mathematical and logical functions. Moreover, a set of functions and operators are provided to combine rules to define new rules. Collection of pre-defined rules can be combined to build specific transformation ruled. Rules are stored in a Transformation Rule Library that supports the automatic generation of JAVA code for rules.

3.3 Step 3: Translator Construction

Once the metamodel and rules are specified, they can be used to construct data model translators. The construction can be carried out in three phases using a tool called "Translator Compiler" [9]. In the first phase, transformation paths are created to join pairs of metatypes in the metamodel. In our example, the transformation path between MRelation and MComplex-Object is [MRelation, MSimple-Object, MComplex-Object]. The construction of transformation paths is based on Dijkstra's algorithm [6].

The second step consists in successive regroupment of transformation paths to create new paths. This is done as follows.

First, transformation paths that possess the same source metatype and the same target metatype are regrouped. In our example the path between MClass and MRelation is constructed from two transformation paths [MClass, MComplex-Object, MSimple-Object, MRelation] and [MClass, MBinary-Link, MRelation]. The path between MIHL and MRelation is [Minheritance-Link, MBinary-Link, MRelation]. This step allows solving problems due to multiple inheritances. The resulting path is [[MClass / Minheritance-Link], [MComplex-Objec / MBinary-Link], [MSimple-Object / MBinary-Link], MRelation].

Next, paths between metatypes generalizing a source model source and metatypes generalizing a target model are grouped. Figure 6 depicts the complete translation from an object OO model (MClass, Minheritance-Link) to a Relational model.

The last phase is the compilation of transformation rules along transformation paths. This compilation generates a specific translator for converting a

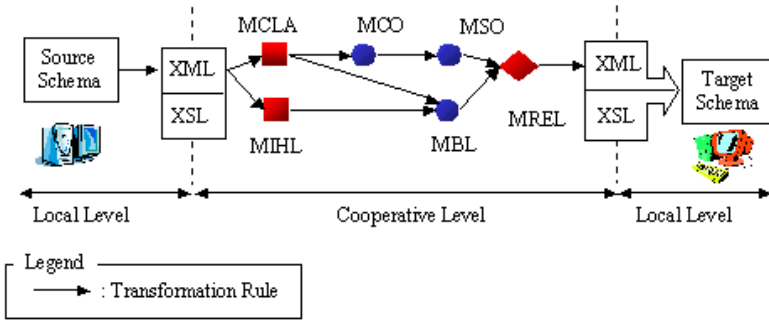


Fig. 6. Transformation paths from Object to Relational metatypes

schema from a source to a target data model and inversely. Java codes associated with rules are combined to generate the translator. The Translator Compiler provides the Java source code of the translator. Corresponding executable files can be created for the various platforms comprising the cooperation.

4 Conclusion

In this paper, we have presented a design methodology and tools framework, called X-TIME, to support the design and management of interoperable information systems. X-TIME allows the construction of specifics metamodels adapted to information systems that participate in cooperation. Moreover, it facilitates the addition of a new information system in the cooperation by providing an XML based schema description tool called X-Editor that can be used to describe local data model concepts. The resulting definitions in XML-DATA is used to update the contents of the metamodel to adapt it to the new cooperation and to define specific translators for new cooperation component. The specialisation hierarchy allows first reusing the definition of every metatype by a mechanism of inheritance and especially to simplify their definition. The underlying metamodel of the X-Time design methodology is characterized by the fact that:

- It provides a minimum set of metatypes that capture the semantics of different categories of concepts found in data models composing the cooperation.
- It achieves extensibility by organizing the metatypes in an specialization hierarchy. Thus, a new metatype is defined by specializing an existing metatype.
- It achieves translation by defining a set of transformation rules, translation paths and translators.
- It allows the reuse of transformation rules and sharing of translation step to reduce the work of translators building.

Our future goal is to extend the above results and to define a formal methodology and algorithm for heterogeneous query processing. This will allow us to define a query interface for the interoperation or migration of existing systems.

References

1. S Abiteboul, P Buneman, and D Suciu. Data on the web. In *From Relations to Semistructured Data and XML*, <http://mkp.com>, 2000. Morgan Kaufmann Publishers.
2. P Atzeni and R Torlone. Mdm : A multiple-data-model tool for the management of heterogeneous database schemes. *Proceedings of the SIGMOD International Conference*, pages 538–531, 1997.
3. T Barsalou and D Gangopadhyay. M(dm) : An open framework for interoperation of multimodel multidatabasesystems. *Proceedings of the 8th International Conference of Data Engineering*, pages 218 – 227, February 1992. Tempe, Arizona.
4. D Bell and J Grimson. Distributed database system. *International Computer Science Series, Addison-Wesley Publishing Company*, 1992.
5. S Cluet, C Delobel, J Siméon, and K Smaga. Your mediator need data conversion! *ACM SIGMOD International Conference on Management of Data*, pages 177–188, June 1998.
6. O J Dahl, E W Dijkstra, and C A R Hoare. Structured programming, second printing. *A.P.E.C. Studies in Data Processing No.8, Academic Press*, 1973. London-New-York.
7. M A Jeusfeld and U A Johnen. An executable metamodel for re-engineering of database schemas. *Proceedings in the 13th International Conference of Entity-Relationship Approach*, 1994. Manchester, UK.
8. L V S Lakshmanan, F Sadri, and I N Subramian. On the logical foundations of schema integration and evolution in heterogeneous database systems. *Deductive and Object-Oriented Database, 3rd International Conference, LNCS 760*, 1993. Phoenix, Arizona.
9. C Nicolle. A translator compiler for interoperable information systems. *17th International CODATA Conference*, October 2000. Baveno, Italy.
10. C Nicolle, D Benslimane, and K Yétongnon. Multi-data models translation in interoperable information systems. In Springer Verlag, editor, *Lecture Notes in Computer Science*, pages 78–89. CAISE, May 1996.
11. M Papazoglou, N Russel, and D Edmond. A translation protocol acheiving consensus of semantics between cooperating heterogeneous database systems. *Proceeding of the First IFCIS International Conference on Cooperative Information Systems*, pages 78–89, june 1996. Brussels, Belgium.
12. A P Sheth and J A Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22, 1990.
13. W3C. Extensible stylesheet language (xsl). <http://www.w3.org/XSL/>, 1998.
14. W3C. Xml-data, w3c note 05 jan 1998. <http://www.w3.org/TR/1998/NOTE-XML-data>, January 1998.
15. W3C. Xml web page. <http://www.w3.org/XML/>, 1998.

E-Speak – An XML Document Interchange Engine

Sven Graupner, Wooyoung Kim, Akhil Sahai, and Dmitry Lenkov

Hewlett-Packard Laboratories

1501 Page Mill Road, Palo Alto, CA 94394, USA

{sven_graupner,wooyoung_kim,akhil_sahai,dmitry_lenkov}@hp.com

<http://www.e-speak.hp.com>

Abstract. E-Speak is Hewlett-Packard's open source infrastructure for web-based e-services that allows e-services to advertise, discover, and interoperate with each other dynamically and securely (www.e-speak.net). It provides XML interfaces for creating, mediating, and accessing e-services in combination with concepts of vocabularies, name virtualization, dynamic discovery, and visibility control.

The paper overviews the E-Speak architecture and its abstractions. Then we describe Web E-Speak, E-Speak's gateway to the Web, and show by examples how Web E-Speak supports the creation, deployment, and discovery of web-based e-services and their interaction.

1 Introduction

E-Speak addresses two important challenges e-services are facing today. First, it is designed to let e-services and intelligent devices communicate and broker services dynamically with one another across the Internet. Any E-Speak enabled e-service can interact with any other regardless of their physical location, platform, system management policy, development environment, or device capability. E-Speak is based on XML, the de facto Internet standard, and leverages Java's platform independence. Furthermore, it is designed with security in mind from the ground up. For example, it mediates all accesses to services and implements secure firewall traversal. The platform and transport independence along with strong security allows disparate entities in the Internet to interact and provide services for each other in a secure, manageable way. Second, E-Speak provides a flexible resource model for service advertisement and a powerful query mechanism for dynamic service discovery. Its dynamic advertisement and discovery abstractions enable e-services in the Internet to interoperate with each other dynamically in very loosely coupled ways.

Web E-Speak is a document interchange engine that routes messages between clients and services through E-Speak. In addition to E-Speak's Java API-based message routing interface, Web E-Speak defines an XML document content model for e-services to access E-Speak's functionality from the Internet, such as advertisement and dynamic discovery. All interactions with Web E-Speak are

modeled as document exchanges. Thus, no programming is required to access E-Speak services. Accessing an E-Speak service is as simple as writing an XML document and sending it to Web E-Speak. The document exchange model abstracts from service implementations and makes interaction language-independent.

Web E-Speak sits in between clients and services and mediates message traffic among them. Mediation is essential for seamless interoperability between clients and services. E-Speak delivers messages to services behind a firewall without disrupting firewall configurations using well-defined and controlled paths through firewalls such as HTTP and Socks. For secure communication, SSL can be used over HTTP between clients and services and Web E-Speak. Stronger end-to-end security is available by E-Speak's session layer security.

2 E-Speak Architecture Overview

E-Speak has a layered architecture where each layer provides a different level of abstractions for e-services' development. At the bottom is the E-Speak engine which provides the basic infrastructure services, such as service registration and advertisement, discovery, security, and management. Above the bottom sit programming libraries which comprise the E-Speak Service Interface (ESI) to the E-Speak engine. Web E-Speak is the interface that supports the document exchange model. E-Speak also defines a service bus that allows e-services to dynamically compose new e-services.

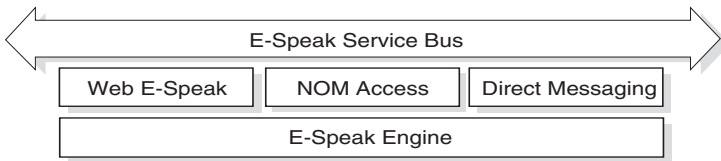


Fig. 1. The E-Speak architecture.

E-Speak builds all abstractions and system functionality on one single, first-class entity - resource [4]. A *resource* is a uniform representation of an entity created in or registered with the E-Speak engine. E-Speak does not distinguish active services, such as name services or printing services, from passive resources, such as files and data entries. Rather, it treats them uniformly by dealing only with their descriptions needed for advertisement and discovery.

Suppose a user creates a file service and registers it with the E-Speak engine. The corresponding file resource within the engine is nothing more than a description of the actual file including name and size, and a specification including the access control policy. The E-Speak engine does not access the file directly. Rather it keeps a mailbox for this resource and routes requests to the file service through its mailbox. The file service defines a handler watching the mailbox for

¹ This *resource* is an E-Speak term and should not be confused with computational resources such as files, processes, etc.

incoming requests and accesses the file upon request. In this context, E-Speak also resembles a runtime system for Actors [16].

The notion of a resource in E-Speak is such general that it uniformly covers computing resources and services as well as e-services used in e-commerce environments. The reason for this powerful generality is that the variety of entities occurring in different service deployment environments follows a pattern: they all can be described by meta-data (as shown in more detail below), and they interact based on messages routed among the entities.

Service providers register a service with an E-Speak engine by submitting a service *specification* and *descriptions*. In response to a registration request, the engine creates a representation of a resource or a service. The descriptions may be advertised to other E-Speak engines. A description is about how a resource or a service is presented to users while a specification defines the access information such as interfaces and the access control policy. The dichotomy of the resource representation allows a flexible, yet secure service discovery framework [3].

E-Speak's attribute-based descriptions lead to potential name collisions. Suppose a 21-inch monitor as well as a 21-inch TV is registered. When a user tries to find a TV using a query "*size=21*" she may unexpectedly find the 21-inch monitor as well. To avoid name collision and to facilitate description and query validation, E-Speak introduces a powerful abstraction called *vocabulary* and requires descriptions be specified in a specific vocabulary. A vocabulary defines a name space which consists of a set of valid attributes and their types. Attribute names are qualified with vocabulary references in order to resolve name ambiguities. The engine validates descriptions and query constraints against vocabularies. This notion of name spaces is similar to name spaces defined by XML schema [2]. Vocabularies naturally partition the search space of descriptions. This allows vocabularies to evolve over time independently of other vocabularies.

E-Speak supports coarse- as well as fine-grain *security*. First, a user may share services with specific users by creating own vocabularies and making them visible only to certain users. Since vocabularies partition the search space of descriptions, those who are not aware of the vocabulary cannot find services.

If requested by a client, E-Speak virtualizes names that identify services. With *name virtualization* neither service providers nor clients need to reveal their identity to interact with each other. The engine protects the mapping information from virtual names to actual services. Combined with dynamic discovery, name virtualization may also be used for dynamic fail-over policies, run-time upgrades, and service relocation without disrupting clients using these services. Furthermore, it may be used to implement transparent load balancing policies replicated services appearing behind the same (virtualized) name.

E-Speak mediates all requests to services. *Mediation* is enforced by attribute certificates based on the Simple Public Key Infrastructure (SPKI). Different access rights to services are granted depending on authenticated attributes. Service access in E-Speak is based on asynchronous messaging. On top of it, E-Speak libraries build more user friendly interaction models such as the Java-based Network Object Model (NOM, refer to Figure 1) and the XML-based document exchange model provided by Web E-Speak.

The E-Speak engine provides the following infrastructure services:

Message routing/mediation. Requests and replies are routed through potentially multiple E-Speak engines. Reliable messaging is supported.

Advertisement/registration. Services may be described in multiple vocabularies. The descriptions may be advertised to other E-Speak engines.

Dynamic discovery. E-Speak provides a powerful querying mechanism to discover services dynamically from other E-Speak engines.

Security/visibility control. E-Speak implements attribute certificate-based security using SPKI to mediate all accesses to services. Name translation can be used to hide identities of service providers and clients.

Firewall traversal. E-Speak engines implement secure inter-engine communication across firewalls using HTTP or Socks V5.

Persistence. E-Speak provides a persistent repository such that resource descriptions are not lost even with unexpected failures of the engine.

Events/management. E-Speak uses a unique publish/subscribe/distribute model for event distribution. It is used by E-Speak's management services.

3 Web E-Speak: A Document Interchange Engine

The Internet's nature of *laissez faire* makes it a daunting challenge for a document interchange engine to achieve seamless interoperability. To mediate communication between heterogeneous clients and services and make them interoperate across the Internet, a document interchange engine needs to address three important issues. First, it should abstract different transport protocols. Incoming messages should be translated into a uniform intermediate representation to simplify implementations and to facilitate message routing. The engine should provide schema translation support to resolve potential schema incompatibilities.

Figure 2 illustrates the architecture of the Web E-Speak document interchange engine. It contains four components. The adapters receive inbound documents and translate them to an intermediate representation. The translators perform schema translation to resolve potential schema inconsistencies. The router delivers a document using an appropriate protocol handler. The protocol handlers implement different transport protocols.

In addition, a document interchange engine may provide other infrastructure services as value-adds such as reliable delivery of asynchronous messages with persistence support. Such services may authenticate users and manage user sessions to protect clients and services from impersonators. Some clients or services may request encryption-based data protection. Other services may provide dynamic advertisement/discovery mechanisms and so forth.

Web E-Speak uses XML over HTTP. It is implemented using the Java servlet interface of standard web servers and follows a pipeline architecture. Requests arrive at one side of the pipeline as HTTP and flow to the other end. Web E-Speak allocates a thread for each request in order to process multiple requests simultaneously. Large scale deployments can be supported by replicating Web

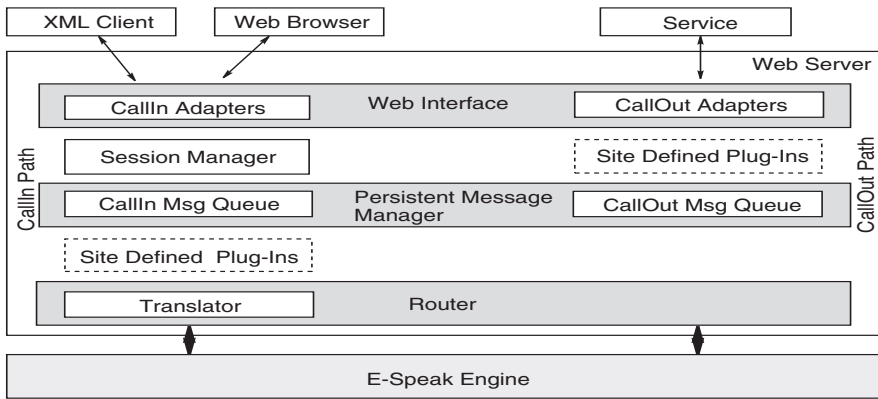


Fig. 2. Web E-Speak architecture with a pluggable pipeline.

E-Speak instances on different server machines. While requests are traveling along the pipeline, each module performs assigned operations. The Web E-Speak pipeline consists of the following, configurable modules as shown in Figure 2.

Web Server. The web server provides HTTP and Java servlet support. Web E-Speak can be used with standard SSL.

Web Interface Adapters. The web interface contains so-called callin and call-out adapters converting incoming messages into internal DOM representations and apply transformations to outgoing messages mapping message content into XML, HTML, or WML.

Session Manager. The session manager controls sessions for clients and services using the account management service of the E-Speak engine. Only authenticated users can use Web E-Speak. Any attempt to access Web E-Speak is denied unless a request contains a valid session token.

Persistent Message Manager. It stores requests and replies in a persistent message queue so that services and clients can pick them up later when they re-connect to Web E-Speak. This e-mail-like, persistent, asynchronous messaging enables loosely coupled interaction among services which are not expected being available and connected all the time.

Router. The router delivers messages internally among E-Speak engines and delivers messages to E-Speak infrastructure services.

Web E-Speak defines schemas and a set of XML elements for routing messages through E-Speak engines. Messages must contain an XML header that conforms to the Web E-Speak message header schema [5]. The header contains routing information. It also has the session information and message identifiers. The header may have an additional description of the message content and processing instructions such as the encryption or the compression method.

If the message's destination is one of E-Speak's infrastructure services, the document following the header must be an XML document conforming to the

Web E-Speak document content model [5]. The document then specifies an operation to be performed as well as arguments to the operation. If the destination is a regular external service, the body is encoded in MIME format and not considered by the E-Speak router. Bodies may contain arbitrary, MIME encoded content, including binary content allowing tunneling any application-level protocol through Web E-Speak.

4 E-Speak Infrastructure Services

The E-Speak engine provides a range of infrastructure services for e-services' collaboration [4]. In this section we describe the most relevant infrastructure services made available through Web E-Speak.

4.1 Advertisement with Service Registration

A service is registered with an E-Speak engine with a specification and descriptions. The specification is composed of interfaces, security policies, and a filter constraint (Section 4.3). Descriptions may be specified in different vocabularies. The `locator` element in the specification contains the actual service address.

```
<?xml version="1.0"?>
<resource xmlns="http://www.e-speak.net/Schema/E-Speak.register.xsd">
  <resourceSpec><locator>mailto:reservation@my.porsche.com</locator></resourceSpec>
  <resourceDes name="car-dealership">
    <vocabulary>Car-Dealership-Vocab</vocabulary>
    <attr name="name"><value>John Doe's Porsche in Palo Alto</value></attr>
    <attr name="car-type"><value>Porsche</value></attr>
    <attr name="contact"><value>555-555-5555</value></attr>
    <attr name="street"><value>555 TheStreet Ave.</value></attr>
    <attr name="city"><value>Palo Alto</value></attr>
    ...
  </resourceDes>
  <resourceDes name="car-repair">
    <vocabulary>Car-Mechanics-Vocab</vocabulary>
    <attr name="name"><value>John Doe's Performance Car Clinic</value></attr>
    <attr name="specialty"><value>Performance Car</value></attr>
    <attr name="contact"><value>555-555-5555</value></attr>
    <attr name="street"><value>555 TheStreet Ave.</value></attr>
    <attr name="city"><value>Palo Alto</value></attr>
    ...
  </resourceDes>
</resource>
```

Fig. 3. A registration request which registers a Porsche dealership which also opens a car repair shop. The car dealership and the repair shop are described in the Car-Dealership-Vocab and Car-Mechanics-Vocab, respectively.

4.2 Dynamic Service Lookup

Services are discovered by constructing and sending queries to the E-Speak engine. A query contains a *constraint*, zero or more *preferences*, and an *arbitration* policy. It may have vocabulary declarations if attributes refer to multiple vocabularies. A constraint specifies a query condition. Preferences are applied to order results. The arbitration policy specifies how many results are returned.

```

<?xml version="1.0"?>
<esquery xmlns="http://www.e-speak.net/Schema/E-Speak.query.xsd" >
  <from src="http://www.john-doe-two.com/" />
  <vocabulary name="v1" src="Car-Dealership-Vocab"/>
  <vocabulary name="v2" src="Car-Mechanics-Vocab"/>
  <result>$serviceInfo</result>
  <where><condition> v1:car-type ='Porsche' and v1:city='Palo Alto'
                    and v2:specialty ='Performance Car' and v2:city='Palo Alto'
  </condition></where>
  <arbitration><cardinality>all</cardinality></arbitration>
</esquery>

```

Fig. 4. A lookup request where a user is trying to find a Porsche dealership in Palo Alto which also has a car repair shop for performance cars in the same city.

A vocabulary declaration associates a local vocabulary reference to a vocabulary. Local references are then used to distinguish between attributes from different vocabularies (': ' operator). A vocabulary declaration is specified with a **vocabulary** element which contains a **name** attribute for a local reference and a **src** attribute for a vocabulary. In Figure 4, the vocabulary names **v1** and **v2** are used to qualify vocabularies. Constraints are specified with the **where** element. Identifiers in constraints, such as **specialty** or **city**, then refer to attributes.

When the E-Speak engine finds matching descriptions, it uses preferences to order them. Preferences come with three flavors. The **min** and **max** operators order services in ascending or descending order. Using the **with** operator, multiple preferences can be prioritized (see [3] for the precise semantics).

Arbitration limits the number of returned matches. A cardinality of a positive integer **n** requests the engine to return at most **n** services. A client may ask to return **all** services or **any** (one randomly selected) service from the set of matches.

4.3 Visibility Control through Filtering

Service providers may want to control the visibility of their services to clients. A mortgage broker may want clients with good credit history to find mortgage programs with preferred interest rates, for example. A chip design company may want only its chip design engineers to find very high-resolution plotters and printers and not other employees. Visibility control is specified with a filter constraint at registration time (see Figure 5). A filter constraint is a predicate over attributes of a service and may also refer to user profile information. A filter constraint is evaluated when a service matches a user's query. Only those services whose filter constraint evaluates to true are included in the result set. A meta variable **\$user** is used in the example to refer to the profile information of a user who requests the query.

```

<vocabulary name="addr" src="Address" />
<filter>$user/addr:state='CA' or $user/addr:state='California'</filter>

```

Fig. 5. A filter constraint. Only users located in California will find services.

4.4 Dynamic Attribute

Most attributes remain static after services are registered unless they are explicitly modified. However, some attributes are dependent on dynamically changing attributes of other resources. These dynamic attributes need to be computed at query time. For example, a US company may want to allow customers in Europe to find products based on the price marked in euro. The price depends on the euro/dollar exchange rate at the time of the lookup.

```
<variable name="x">query for the euro/dollar exchange rate</variable>
<attrDecl name="price_in_euro" required="true">
  <datatypeRef name="float" dynamic="true"><value>price_in_dollar*$x</value></datatypeRef>
</attr>
```

Fig. 6. A dynamic attribute. When a query is given which refers to `price_in_euro`, the engine finds the exchange rate and computes the price.

4.5 Vocabulary Creation

Vocabularies themselves are resources managed by the E-Speak engine. As resources, vocabularies should be described in vocabularies as well. Creation of a vocabulary requires another vocabulary which requires another, etc. To end the recursion, E-Speak defines a *base vocabulary* which is a priori defined across all E-Speak engines. The attributes of the base vocabulary are predefined and include `name`, `type`, and `version`.

Anybody can create and register a vocabulary. Creating and registering a vocabulary is essentially the same as the registration of other services. The only difference is that a vocabulary creation requires a vocabulary definition (Figure 7). A vocabulary definition is a set of attribute specifications which define the name and the data types of the attributes comprising the vocabulary to be created. Since anyone can create vocabularies, equal vocabularies may co-exist.

```
<?xml version="1.0"?>
<resource xmlns="http://www.e-speak.net/Schema/E-Speak.register.xsd">
  <resourceDes>
    <vocabulary>http://www.e-speak.net/Schema/E-Speak.base.xsd</vocabulary>
    <attr name="Name"><value>used-car</value></attr>
    <attr name="Type"><value>Vocabulary</value></attr>
  </resourceDes>
  <attrGroup name="used-car" xmlns="http://www.e-speak.net/Schema/E-Speak.vocab.xsd">
    <attrDecl name="make" required="true"><datatypeRef name="string"/></attrDecl>
    <attrDecl name="model" required="true"><datatypeRef name="string"/></attrDecl>
    <attrDecl name="ask-price" required="true">
      <datatypeRef name="float"><default>0.0</default>
      <minInclusive>0.0</minInclusive></datatypeRef>
    </attrDecl>
    ...
  </attrGroup>
</resource>
```

Fig. 7. A vocabulary creation request which creates a vocabulary named `used-car` with attributes `make`, `model`, and `ask-price`.

5 Related Works

The simplicity and interoperability of XML has encouraged companies to collaborate in developing XML-based frameworks and protocols for automatic information exchange and business execution. E-Speak has been developed with platform independent technologies such as Java, XML, and TCP/IP as well as SOAP/HTTP. It provides a flexible and secure communication infrastructure with which following frameworks can co-operate (see <http://www.e-speak.hp.com/product/comparison.shtml>).

- **BizTalk.** Microsoft's BizTalk Framework is an XML framework for application integration and electronic commerce which consists of a specification for constructing XML documents, and XML schemas for message routing. It achieves interoperability between heterogeneous applications by decoupling issues like document validation and message routing from applications (<http://www.biztalk.org>). We can make E-Speak interoperate with the BizTalk server by translating BizTalk message headers to E-Speak message headers, applicable for functions available in both systems.
- **eCo.** CommerceNet's eCo framework aims to business-level integration that enables companies to publish interfaces and to discover other companies' interfaces on the Web and determine how they can do business with them (<http://www.commerce.net>).
- **RosettaNet.** RosettaNet is a consortium of companies in the Information Technology (IT) supply chain sector. It has created a common framework and language that all partners of an IT supply chain implement in order to automate their business interactions. The framework separates Partner Interface Processes (PIPs) from implementation details such as how to model trading partner agreements between businesses and what public key infrastructure (PKI) partners will use, as well as from communication details such as message packaging and message sequencing (<http://www.rosettanet.org>).
- **Universal Description, Discovery and Integration (UDDI).** UDDI is a project initiated by Ariba, IBM, and Microsoft to create a platform independent framework for describing businesses, discovering business services and integrating them. A public Business Registry has been defined as part of the UDDI effort². It has been being built on Internet standards such as XML and HTTP as well as the Simple Object Access Protocol (SOAP) messaging specification. Businesses register their services in terms of white page information for business contact information, yellow page information for service categorization, and green page information for technical information, such as e-business rules, service descriptions, application invocation, and data binding (<http://www.uddi.org>). E-Speak provides more flexible and richer match-making mechanisms with vocabularies.

² HP recently announced to host a public Business Registry.

- **Web Services Description Language (WSDL).** WSDL proposed by IBM and Microsoft is an XML format for describing the interface, protocol bindings and the deployment details of network services. Using WSDL, service providers describe network services as a set of endpoints capable of exchanging messages. Messages and operations are defined separately and bound to a concrete network protocol and message format of a particular endpoint. Related endpoints are combined into services. Using WSDL together with E-Speak offers a secure communication medium for the interaction between services (<http://www.w3.org/TR/wsdl>).

6 Summary

E-Speak, HP's strategic technology for e-services, provides a platform for seamless service interaction, advertisement and discovery in the Internet. Web E-Speak provides the gateway to the Web using XML to integrate heterogeneous e-services and smart devices.

The paper describes the E-Speak architecture and functionality. Basic concepts of vocabularies, dynamic discovery, name virtualization, visibility control, and mediation are explained. Then, the architecture of Web E-Speak is discussed, and examples show the interaction between e-services and Web E-Speak. Finally, E-Speak is compared with other systems and frameworks in the e-services arena.

E-Speak A.0 has been released as a product in January 2001 on which the paper is mostly based on. The examples were taken from E-Speak Beta 3.0, released in May 2000 under the GNU GPL license. The software, source code, documents, and other related information is available at the open source web site <http://www.e-speak.net> or at the HP corporate site <http://www.e-speak.hp.com>.

References

1. G. Agha, S. Frølund, W. Kim, R. Panwar, A. Patterson, and D. Sturman. Abstraction and Modularity Mechanisms for Concurrent Computing. *IEEE Parallel and Distributed Technology: Systems and Applications*, 1(2):3–14, May 1993.
2. World Wide Web Consortium. XML Schema Part 1: Structures, April 2000. W3C Working Draft. <http://www.w3c.org/TR/xmlschema-1/>.
3. S. Graupner, W. Kim, D. Lenkov, and A. Sahai. E-Speak: an Enabling Infrastructure for Web-based E-services. In *Proceedings of International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet (SSGRR 2000)*, ISBN 88-85280-52-8, l'Aquila, Italy, August 2000.
4. Hewlett-Packard. E-Speak Architectural Specification. Developer Release 3.01, June 2000. <http://www.e-speak.net/library/pdfs/E-speakArch.pdf>.
5. Hewlett-Packard. E-Speak WebAccess Programmer's Guide. Developer Release 3.0, May 2000.
6. W. Kim and G. Agha. Efficient Support of Location Transparency in Concurrent Object Oriented Programming Languages. In *Proceedings of Supercomputing '95*, 1995.

Feature Matrices: A Model for Efficient and Anonymous Web Usage Mining

Cyrus Shahabi, Farnoush Banaei-Kashani, Javed Faruque, and Adil Faisal

Department of Computer Science, Integrated Media Systems Center, University of
Southern California, Los Angeles, CA 90089-2561, USA
[shahabi,banaeika,faruque,faisal]@usc.edu

Abstract. Recent growth of startup companies in the area of Web Usage Mining is a strong indication of the effectiveness of this data in understanding user behaviors. However, the approach taken by industry towards Web Usage Mining is off-line and hence intrusive, static, and cannot differentiate between various roles a single user might play. Towards this end, several researchers studied probabilistic and distance-based models to summarize the collected data and maintain only the important features for analysis. The proposed models are either not flexible to trade-off accuracy for performance per application requirements, or not adaptable in real-time due to high complexity of updating the model. In this paper, we propose a new model, the FM model, which is flexible, tunable, adaptable, and can be used for both anonymous and on-line analysis. Also, we introduce a novel similarity measure for accurate comparison among FM models of navigation paths or cluster of paths. We conducted several experiments to evaluate and verify the FM model.

1 Introduction

Understanding and modeling user behaviors by analyzing users' interactions with digital environments, such as Web-sites, is a significant topic that has resulted in vast recent commercial interests. Commercial products such as *Personify*TM [HREF1], *WebSideStory*TM [HREF2], and *WebTrends*TM [HREF3], and acquired companies such as *Matchlogic*TM, *Trivida*TM, and *DataSage*TM are all witnesses of such interests. In addition, several researchers in various industrial and academic research centers are focusing on this topic [1][3][4][5][7]. A nice survey is provided by Srivastava et al. [8]. Meaningful interpretation of the users' digital behavior is necessary in the disparate fields of e-commerce, distance education, online entertainment and management for capturing individual and collective profiles of customers, learners and employees; for targeting customized/personalized commercials or information, and for evaluating the information architecture of the site by detecting the bottlenecks in the information space.

Understanding user behavior from Web usage data, a.k.a Web Usage Mining (WUM), involves the following three steps: 1) collection and preparation of usage data, 2) pattern discovery from usage data, and 3) personalization or

recommendation based on discovered patterns. One typical but naive approach taken by industry is to first build a static profile for each user based on his/her past navigations during the first and second steps. Subsequently, whenever the user re-visits the site, recommend or personalize contents based on the user profile. There are several drawbacks with this approach. First, for this approach to work, the user identity should be detected using some intrusive technique such as through cookies. Second, the profile cannot distinguish between different roles the user might play during various navigations. For example, buying rock music CDs as a gift for a friend versus purchasing classical music CDs for oneself. Third, if multiple users share the same computer/client, the profile becomes a mixture of several (possibly conflicting) tastes. For example, some of us have experienced *amazon.com* suggesting to us those books that are favorites of our significant others. Finally, the profile becomes static and cannot capture changes in the user taste.

To address these problems, we proposed statistical approaches where first a profile is built for a collection of users with similar navigation paths [10]. Then, recommendations/personalizations are mapped to these profiles through some learning mechanism (e.g., by an expert or via utilization of training data). Finally, the navigation of a new user is compared to different profiles and the recommendations/personalizations will be based on either the closest profile (hard classification) or some weighted aggregate of several profiles (soft classification). Therefore, the reaction to a user is not based on his/her previous actions but based on his/her recent activities. Although this approach, henceforth denoted as “*anonymous WUM*”, eliminates the problems with the naive approach, it imposes new challenges to the design of all the three steps of WUM.

In this paper, we propose a novel model, *Feature Matrices* (FM), that not only addresses the new design challenges of anonymous mining, but also can be used for better conventional WUM. For anonymous WUM, during the first step every single action of a new user should be accurately tracked or else the system cannot react to the user in a timely manner. However, due to the large volume of navigation data generated per site, even if this data can be logged accurately online, it cannot be analyzed real-time unless some aspects of the data be dropped out. Hence, in order for a model to be used real-time, it should be flexible to capture less or more features of the logged data in real-time depending on the volume of navigation. During the second step, user clusters must be generated out of several navigation paths. A model is needed to conceptualize each user navigation path and/or a cluster of user paths. The model should be tunable so that one can trade performance for accuracy or vice-versa. Also, in order for the model to be adaptive, it should be possible to update it incrementally as a new path become available. Finally, the third step deals with personalizing the content or recommending new contents towards the user preferences by comparing a new user *partial* navigation path to the model. This process should be performed efficiently enough before the user leaves the site.

The FM model is in fact a generalization of the Vector model proposed by Yan et al. [9]. It is flexible to allow striking a compromise between accuracy

and efficiency, given the specific requirements of an application. Meanwhile, the FM model can be updated both off-line and, incrementally, online so that it can adapt to both short-term and long-term changes in user behaviors. The FM model benefits from two types of flexibility. First, it can be tuned to capture both spatial and temporal features of Web usage data. In addition, it is an open model so that new features can be incorporated as necessary by an application domain. Second, it has the concept of *order*, similar to that of the Markov model [1], which can be increased to capture more data about the various features of navigations.

Another contribution of this paper is proposing a novel similarity measure, *PPED*, for comparing the FM models of partial paths with clusters of paths. With anonymous WUM, it is critical to accurately compare a partial navigation path of a new user to the cluster representatives. Moreover, we investigate a dynamic clustering approach used to make the system adaptable to short-term changes in user behaviors. Although the dynamic clustering can be executed real-time and incrementally, its accuracy is only 10% worse than that of K-Means. It is important to note that we are not proposing to replace periodical re-clustering with dynamic clustering. Instead we are advocating a hybrid approach. A thorough experimental study is conducted that evaluates and verifies the FM model.

The remainder of this paper is organized as follows. In Section 2, we formally define the FM model. Section 3 explains our novel similarity measure. In Section 4, we discuss our dynamic clustering technique. The results of our experiments are included in Section 5. Finally, Section 6 concludes the paper.

2 The Feature Matrices Model

Here, we present a novel model to represent both sessions and clusters in the context of WUM. We denote this model as the *Feature Matrices (FM)* model. With FM, features are indicators of the information embedded in sessions. In order to quantify the features, we consider universal set of segments in a concept space as basis for the session space. Thus, features of a session are modeled and captured in terms of features of its building segments. This conceptualization is analogous to the definition of basis for a vector space, i.e. “a set of linearly independent vectors that construct the vector space”. Therefore, the FM model allows analyzing sessions by analyzing features of their corresponding segments.

For the remainder of this section, we explain and analyze the FM model. First, we define our terminology. Next, basics of the FM model are explained: the features captured from user interactions, and the main data structure used to present these features. Subsequently, we discuss how to extract the session FM model and the cluster FM model, separately. Finally, we analyze complexity and completeness of the model.

2.1 Terminology

Web-site. A Web-site can be modeled as a finite set of static and/or dynamic Web pages.

Concept Space (Concept). Each Web-site, depending on its application, provides information about one or more concepts. For example, *amazon.com* includes concepts such as *Books*, *Music*, *Video*, etc. The Web pages within a Web-site can be categorized based on the concept(s) to which they belong. A *concept space* or simply *concept* in a Web-site is defined as the set of Web pages that contain information about a certain concept. Note that contents of a Web page may address more than one concept, therefore concept spaces of a Web-site are not necessarily disjoint sets.

Path. A *path* P in a Web-site is a finite or infinite sequence of pages:

$$x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_i \rightarrow \dots \rightarrow x_s$$

where x_i is a page belonging to the Web-site. Pages visited in a path are not necessarily distinct.

Path Feature (Feature). Any spatial or temporal attribute of a path is termed a *path feature* or *feature*. Number of times a page has been accessed, time spent on viewing a page, and spatial position of a page in the path are examples of features.

Session. The path traversed by a user while navigating a concept space is considered a *session*. Whenever a navigation leaves a concept space (by entering a page that is not a member of the current concept), the session is considered to be terminated. Since each page may belong to more than one concept, several *sessions* from different concepts may be embedded in a single *path*. Also, several sessions from the same concept may happen along a path, while user leaves and then re-enters the concept. For analysis, we compare sessions from the same concept space with each other. Distinction between the “session” and the “path” notions makes the comparison more efficacious. To identify the user behavior, we can analyze all the sessions embedded in his/her navigation path, or prioritize the concepts and perform the analysis on the sessions belonging to the higher priority concept(s). Moreover, among the sessions belonging to the same concept space, we can restrict our analysis to the longer session(s), to decrease complexity of the analysis based on the application specifications. In any case, the result of the analysis on different sessions of the same path can be integrated to provide the final result. For example, in a recommendation system, the recommendation can be generated based on various user preferences detected by analyzing different sessions of the user’s navigation path. Thus, hereafter we assume all sessions belong to the same concept. Similar analysis can be applied to sessions in any concept space.

Session Space. The set of all possible sessions in a concept space is termed *session space*.

Path Segment (Segment). A *path segment* or simply *segment* E is an n -tuple of pages: $(x_1, x_2, \dots, x_i, \dots, x_n)$. We denote the value n , as the *order* of the segment E ($n \geq 1$). Note that there is a one-to-one correspondence between tuples and

sequences of pages; i.e. $(x_1, x_2, \dots, x_i, \dots, x_n) \equiv x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_i \rightarrow \dots \rightarrow x_n$. We use tuple representation because it simplifies our discussion. Any subsequence of pages in a path can be considered as a segment of the path. For example, the path $x_1 \rightarrow x_3 \rightarrow x_2 \rightarrow x_5 \rightarrow x_2$ contains several segments such as 1st order segment (x_1) , 2nd order segment (x_3, x_2) , and 4th order segment (x_3, x_2, x_5, x_2) . We exploit the notion of segment as the building block of sessions in order to model their features.

Universal Set of Segments. $\varepsilon_C^{(n)}$, universal set of order- n segments, is the set of all possible n -tuple segments in the concept space C . Hereafter, since we focus on analysis within a single concept, we drop the subscript C from the notation.

Cluster. A *cluster* is defined as a set of similar sessions. The similarity is measured quantitatively based on an appropriate similarity measure (see Section 3).

2.2 Basics

Features. We characterize sessions through the following features:

- *Hit (H):* Hit is a spatial feature that reflects which pages are visited during a session. The FM model captures H by recording the number of times each *segment* is encountered in a traversal of the session. Reader may consider H as a generalization of the conventional “hit-count” notion. Hit-count counts number of hits per *page*, which is a segment of order 1.
- *Sequence (S):* Sequence is an approximation for the relative location of pages traversed in a session. As compared to H , it is a spatial feature that reflects the location of visits instead of the frequency of visits. With the FM model, S is captured by recording relative location of each segment in the sequence of segments that construct the session. If a segment has been repeatedly visited in a session, S is approximated by aggregating the relative positions of all occurrences. Thus, S does not capture the exact sequence of segments. Exact sequences can be captured through higher orders of H .
- *View Time (T):* View time captures the time spent on each segment while traversing a session. As opposed to H and S , T is a temporal feature.

Features of each session are captured in terms of features of the segments within the session. We may apply various orders of universal sets as basis to capture different features. Throughout our discussion, we have used $\varepsilon^{(1)}$ for T , and $\varepsilon^{(2)}$ for H and S , unless otherwise stated. Therefore, we extract the feature T for single-page segments, x_i , and features H and S for ordered page-pair segments (x_i, x_j) . In Section 2, we will explain how using higher order bases results in more complete characterization of the session by the FM model in expense of higher complexity.

The FM model is an open model. It is capable of capturing any other meaningful session features in addition to those mentioned above. The same data structure can be employed to capture the new features. This is another option with which completeness of the FM model can be enhanced. However, our

experiments demonstrate that the combination of our proposed features is comprehensive enough to detect the similarities and dissimilarities among sessions appropriately.

Data Structure. Suppose $\varepsilon^{(n)}$ is the basis to capture a feature F for session U , we deploy an n -dimensional *feature matrix*, $M_{r^n}^F$, to record the F feature values for all order- n segments of U . n -dimensional matrix M_{r^n} is a generalization of 2-dimensional square matrix $M_{r \times r}$. Each dimension of M_{r^n} has r rows, where r is the cardinality of the concept space. For example, $M_{4 \times 4 \times 4}$ that is a cube with 4 rows in each of its 3 dimensions, is a feature matrix for a 4-page concept space with ε^3 as the basis. Dimensions of the matrix are assumed to be in a predefined order. The value of F for each order- n segment $(x_\alpha, x_\beta, \dots, x_\omega)$ is recorded in element $a_{\alpha\beta\dots\omega}$ of $M_{r^n}^F$. To simplify the understanding of this structure, reader may assume that rows in all dimensions of the matrix are indexed by a unique order of the concept space pages; then the feature value for the order- n segment $(x_\alpha, x_\beta, \dots, x_\omega)$ is located at the intersection of row x_α on the 1st dimension, row x_β on the 2nd dimension, ... , and row x_ω on the n -th dimension of the feature matrix. Note that M_{r^n} covers all order- n segment members of $\varepsilon^{(n)}$. for instance, in a 100-page concept space with $\varepsilon^{(2)}$ as the basis, M_{100^2} has 10000 elements. On the other hand, number of segments existing in a session usually is in the order of tens. Therefore, M_{r^n} is usually a sparse matrix. The elements for which there is no corresponding segment in the session are set to zero.

To map a session to its equivalent FM model, the appropriate feature matrices are extracted for features of the session. The entire set of feature matrices generated for a session constitutes its FM model:

$$U^{fm} = \{M_{r^{n_1}}^{F_1}, M_{r^{n_2}}^{F_2}, \dots, M_{r^{n_m}}^{F_m}\}$$

If $n = \max(n_1, n_2, \dots, n_m)$ then U^{fm} is an order- n FM model.

In subsequent sections, we explain how values of different features are derived for each segment from the original session, and how they are aggregated to construct the cluster model.

2.3 Session Model

Here, we explain how values of different features are extracted from a session to form the feature matrices of its FM model. Recall that we record features of a session in terms of features of its segments. Thus, it suffices if we explain how to extract various features for a sample segment E :

- For Hit (H), we count the number of times E has occurred in the session ($H \geq 0$). Segments may partially overlap. As far as there is at least one non-overlapping page in two segments, the segments are assumed to be distinct. For example, the session $x_1 \rightarrow x_2 \rightarrow x_2 \rightarrow x_2 \rightarrow x_1$, has a total of 4 order-2 segments, including 1 occurrence of (x_1, x_2) , 2 occurrences of (x_2, x_2) , and 1 occurrence of (x_2, x_1) .

- For Sequence (S), we find the relative positions of every occurrence of E and record their arithmetic mean as the value of S for E ($S > 0$). To find the relative positions of segments, we number them sequentially in order of appearance in the session. For example, in the session $x_1 \rightarrow^1 x_2 \rightarrow^2 x_2 \rightarrow^3 x_2 \rightarrow^4 x_1$, S value for the segments (x_1, x_2) , (x_2, x_2) , and (x_2, x_1) are 1, 2.5 ($= \frac{2+3}{2}$), and 4, respectively.
- For View Time (T), we add up the time spent on each occurrence of E in the session ($T \geq 0$).

2.4 Cluster Model

With *clustering*, user sessions are grouped into a set of clusters based on similarity of their features. To cluster sessions, since the FM model is a distance-based model, we need a similarity measure to quantify the similarity between sessions, and a clustering algorithm to construct the clusters. Moreover, we need a scalable model for the cluster. Nowadays, any popular Web-site is visited by a huge number of users. In such a scale, we may employ any similarity measure and clustering algorithm to group the sessions (or better to say session models) into clusters, but mere grouping the sessions is not sufficient. If a cluster is naively modeled as a set of session models, any analysis on a cluster will be dependent on the number of sessions in the cluster which is not a scalable solution. Therefore, we are desperately in need of a cluster model that can be used for analysis independent of the number of cluster members. In this section, we describe our cluster model. Subsequently, in Section 3, we introduce an accurate similarity measure for the purpose of clustering, and finally, in Section 4, we propose a variation to conventional clustering algorithms to make them real-time adaptable to varying behaviors.

With our approach of modeling a cluster, we aggregate feature values of all clustered sessions into corresponding feature values of a virtual session. This virtual session is considered as a representative of all the sessions in the cluster, or equally as the model of the cluster. Consequently, the complexity of any analysis on a cluster will become independent of the cluster cardinality.

Suppose we have mapped all the sessions belonging to a cluster into their equivalent session models. In order to aggregate the features of the sessions into the corresponding features of the cluster model, it is sufficient to aggregate features for each basis segment. Assume we denote the value of a feature F for any segment E in the basis by $F(E)$. We apply a simple aggregation function, namely *arithmetic averaging*, to $F(E)$ values in all sessions of a cluster to find the aggregated value of $F(E)$ for the cluster model. Thus, if M^F is the feature matrix for feature F of the cluster model, and M_i^F is the feature matrix for feature F of the i -th session in the cluster, each element of M^F is computed by aggregating corresponding elements of all M_i^F matrices. This procedure is repeated for every feature of the FM model. The final result of the aggregation is a set of aggregated feature matrices that constitute the FM model of the cluster:

$$C^{fm} = \{M^{F_1}, M^{F_2}, \dots, M^{F_n}\}$$

Therefore, the FM model can uniquely model both sessions and clusters.

As mentioned before, the aggregation function we use for all features is the simple arithmetic averaging function. In matrix notation, the aggregated feature matrix for every feature F of the cluster model C^{fm} is computed as follows:

$$M^F = \frac{1}{N} \sum_{i=1}^N M_i^F$$

where N is the cardinality of the cluster C . The same aggregation function can be applied incrementally, when cluster model has already been created and we want to update it as soon as a new session, U_j , joins the cluster:

$$M^F \leftarrow \frac{1}{N+1} (N \times M^F + M_j^F)$$

This property is termed *dynamic clustering*. In Section 4 we leverage on this property to modify the conventional clustering algorithms to become real-time and adaptive.

2.5 Analysis of the Model

WUM involves three categories of tasks: constructing clusters of sessions (clustering), comparing sessions with clusters, and integrating sessions into clusters. Regardless of the model employed for analysis and the algorithm used for clustering, complexity of constructing the clusters is dependent on N , which is the number of sessions to be clustered. This is true simply because during clustering each session should be analyzed at least once to detect how it relates to other sessions. The FM cluster model is defined so that it reduces the time complexity of the other two tasks. If the complexity of comparing a session with a cluster and integrating it into the cluster is independent of the cluster cardinality, user classification and cluster updating can be fulfilled in real-time.

The price we have to pay to achieve lower space and time complexity is to sacrifice *completeness*¹. If the cluster model is merely the set of member sessions stored in their complete form, although the model is *complete* in representing the member sessions, it does not scale. On the other hand, if we aggregate member sessions to construct the cluster model, the model will lose its capability to represent its members with perfect accuracy. The more extensive aggregation is applied, the less complete the cluster model. The FM model is flexible in balancing this trade-off based on the specific application requirements².

¹ A model is more complete if it is a better approximation for the real session/cluster.

² A formal proof for uniqueness of the FM model for a session/cluster is included in [11].

FM Complexity versus the Vector and Markov Models. Let $FM^{(n)}$ be an FM model of the order n (see Table 1 for the definitions of terms), where $n = \max(n_1, n_2, \dots, n_m)$. In the worst case, $FM^{(n)}$ comprises m n -dimensional matrices $M_{r,n}$, one for each of the model features. Thus, *space* cost of $FM^{(n)}$ is $O(mr^n)$. *Time* complexity for user classification is $O(mL)$ and for updating a cluster by assigning a new session to the cluster is $O(mr^n)$. Therefore, space and time complexity of $FM^{(n)}$ model are both independent of M .

Table 1. Parameters

Parameter	Definition
F_i	i -th feature captured in FM
n_i	Order of the basis used to capture F_i
m	Number of features captured in FM
n	$\max(n_1, n_2, \dots, n_m)$
r	Cardinality of the concept space
L	Average length of sessions
M	Average cardinality of clusters

From $O(mr^n)$, complexity increases exponentially with n , which is the order of the FM model. Based on Property 1, as the order n increases, the FM model becomes more complete in describing its corresponding session or cluster:

Property 1. If $p_1 > p_2$ then $FM^{(p_1)}$ is more complete than $FM^{(p_2)}$

Thus, added complexity is the price for a more accurate model. An appropriate order should be selected based on the accuracy requirements of the specific application. Formal proof for Property 1 is included in [11].

The other crucial parameter in $O(mr^n)$ is m , the number of features captured by the FM model. Features are attributes of the sessions, used as the basis for comparison. The relative importance of these attributes in comparing the sessions is absolutely application-dependent. The FM model is an open model in a sense that its structure allows incorporating new features as the need arises for different applications. Performing comparisons based on more features result in more accurate clustering, though again the complexity is increased.

Now let us compare the performance of FM with two other conventional models, namely the Vector model and the Markov model. The Vector model can be considered as one special case of the FM model. As used in [9], the Vector model is equivalent to an $FM^{(1)}$ model with H as the only captured feature. Thus, the Vector model scales as $O(r)$, but as discussed above, since it is an order-1 FM model, it performs poorly in capturing information about sessions. Our experiments illustrate that an $FM^{(2)}$ model with S and H as its features outperforms the Vector model in accuracy (see Section 5). The other model, typically employed in dependency-based approaches, is the “Markov” model. Although whether or not Web navigation is a Markovian behavior has been the

subject of much controversy [2], the Markov model has demonstrated acceptable performance in the context of WUM [1]. The transition matrix of an order- n Markov model is extractable from H feature matrix of an $FM^{(n+1)}$ model. Thus, the FM model at least captures the same amount of information as with an equivalent Markov model. They also benefit from the same time complexity of $O(L)$ for dynamic user classification. However, the Markov model cannot be updated in real-time because the complexity of updating a cluster is dependent on the cardinality of the cluster. Moreover, the Markov model is not an *open* model, as described for FM because it is defined to capture order and hit.

3 Similarity Measure

A *similarity measure* is a metric that quantifies the notion of “similarity”. To capture behaviors of the Web-site users, user sessions are to be grouped into clusters, such that each cluster is composed of “similar” sessions. Similarity is an application-dependent concept and in a distance-based model such as FM, a domain expert should encode a specific definition of similarity into a pseudo-distance metric that allows the evaluation of the similarity among the modeled objects. With the FM model, these distance metrics, termed *similarity measures*, are used to impose order of similarity upon user sessions. Sorting user sessions based on the similarity is the basis for clustering the users. Some similarity measures are defined to be indicator of dissimilarity instead of similarity. For the purpose of clustering, both approaches are applicable.

In [6], we introduce a similarity measure for session analysis that does not satisfy an important precondition: “the basis segments used to measure the similarity among sessions must be orthogonal”. Here, we define a new similarity measure, *PPED*, particularly defined to alleviate the overestimation problem attributed to pure Euclidean distance measure in the context of WUM [9]. This measure satisfies the mentioned precondition. Before defining the function of this similarity measure, let us explain how the FM model is interpreted by a similarity measure.

With a similarity measure, each feature matrix of FM is considered as a uni-dimensional matrix. To illustrate, assume all rows of an n -dimensional feature matrix are concatenated in a predetermined order of dimensions and rows. The result will be a uni-dimensional ordered list of feature values. This ordered list is considered as a vector of feature values in $R^{(r^n)}$, where r is the cardinality of the concept space. Now suppose we want to measure the quantitative dissimilarity between the two sessions U_1^{fm} and U_2^{fm} , assuming that the certain similarity measure used is an indicator of dissimilarity (analogous procedure applies when the similarity measure expresses similarity instead of dissimilarity). Each session model comprises a series of feature vectors, one for each feature captured by the FM model. For each feature F_i , the similarity measure is applied on the two F_i feature vectors of U_1^{fm} and U_2^{fm} to compute their dissimilarity, D^{F_i} . Since the dissimilarity between U_1^{fm} and U_2^{fm} must be based on all the FM features, the

total dissimilarity is computed as the weighted average of dissimilarities for all features:

$$D^F = \sum_{i=1}^m w_i \times D^{F_i} \quad \left(\sum_{i=1}^m w_i = 1 \right) \quad (1)$$

where m is the number of features in the FM model. D^F can be applied in both hard and soft assignment of sessions to clusters. Weight factor w_i is application-dependent and is determined based on the relative importance and efficacy of features as similarity indicators. In Section 5, we report on the results of our experiments in finding the compromised set of weight factors for H and S features.

Here, we explain our new similarity measure. Throughout this discussion, assume \vec{A} and \vec{B} are feature vectors equivalent to n -dimensional feature matrices M_1^F and M_2^F , and a_i and b_i are their i -th elements, respectively. Vectors are assumed to have $N = r^n$ elements, where r is the cardinality of the concept space.

Projected Pure Euclidean Distance (PPED). *PPED* is a variant of Pure Euclidean Distance measure (*PED*) to alleviate the *overestimation* problem. To illustrate the overestimation problem with *PED*, suppose a user navigates the session U that belongs to cluster C . It is not necessarily the case that the user traverses every segment as captured by C^{fm} . In fact, in most cases user navigates a path similar to only a subset of the access pattern represented by C^{fm} and not the entire pattern. In evaluating the similarity between U^{fm} and C^{fm} , we should avoid comparing them on that part of the access pattern not covered by U or else their dissimilarity will be overestimated. Overestimation of dissimilarity occasionally results in failure to classify a session to the most appropriate cluster.

Assume \vec{A} and \vec{B} are two feature vectors of the same type belonging to a session and a cluster model, respectively. To estimate the dissimilarity between \vec{A} and \vec{B} , *PPED* computes pure Euclidean distance between \vec{A} and the projection of \vec{B} on those coordinate planes at which \vec{A} has non-zero components:

$$PPED(\vec{A}, \vec{B}) = \left(\sum_{i=1, a_i \neq 0}^N (a_i - b_i)^2 \right)^{\frac{1}{2}} \quad (2)$$

where $PPED \in [0, \infty)$. Note that *PPED* is not commutative.

Non-zero components of \vec{A} belong to those segments that exist in the session. Zero values, on the other hand, are related to the remainder of the segments in the basis universal set. By contrasting \vec{A} with the projected \vec{B} , we compare the session and the cluster based on just the segments that exist in the session and not on the entire basis. Thus, the part of the cluster not covered in the session is excluded from the comparison to avoid overestimation.

Since *PPED* can compare sessions with different lengths, it is an attractive measure for real-time clustering where only a portion of a session is available at

any given time (see Section 4). *PPED* also helps in reducing the time complexity of the similarity measurement. According to Equation 2, the time complexity of *PPED* is $O(mL)$ (refer to Table 1 for the definitions of the terms). In Section 5, we report on the superiority of *PPED* performance as compared to two classical similarity measures, i.e. *PED* and cosine of the angle formed by the feature vectors (Vector Angle or *VA*).

4 Dynamic Clustering

As discussed in Section 2, since the FM model of a cluster is independent of the cluster cardinality, any cluster manipulation with FM has a reasonably low complexity. Leveraging on this property, we can apply the FM model in real-time applications.

One benefit of this property is that FM clusters can be updated dynamically and in real-time. Note that in most common cluster representations, complexity of adding a new session to a cluster is dependent on the cardinality of the cluster. Therefore, practically in large scale systems, they are not capable of updating the clusters dynamically. By exploiting *dynamic clustering*, the WUM system can adapt itself to changes in users' behaviors in real-time. New clusters can be generated dynamically and existing clusters adapt themselves to the changes in users' tendencies. Delay-sensitive environments such as stock market, are among those applications for which this property is most advantageous. Figure 1 depicts a simple procedure to perform dynamic clustering when a new session is captured.

```

1. Find the distance/similarity between the session and every cluster available in the
   current cluster set using any reasonable similarity measure;

   // All similarity measures discussed in this paper are applicable. These similarity
   // measures are defined based on the data structure of the FM model

2. If there is no cluster closer than TDC to the session
   create a new cluster and use the FM model of the new session as the cluster model;
   else
       update the closest cluster to the session by joining the session to that cluster;

   // TDC is a threshold value specific to Dynamic Clustering. If the distance between the
   // new session and every existing cluster is more than TDC, then it is reasonable to
   // create a new cluster because a new user behavior has been discovered

```

Fig. 1. An algorithm for *dynamic clustering*

Periodical re-clustering is the typical approach in updating the clusters. This approach results in high accuracy, but it cannot be performed in real-time. According to our experiments to compare the accuracy of the dynamic clustering with that of a periodical re-clustering (see Section 5), dynamic clustering shows

lower accuracy in updating the cluster set. In fact, with dynamic clustering, we are trading accuracy for adaptability. Thus, dynamic clustering should not be used instead of classical clustering algorithms, but a hybrid solution is appreciated. That is, the cluster set should be updated in longer periods through periodical re-clustering to avoid divergence of the cluster set from the trends of the real user behaviors. Meanwhile, dynamic clustering can be applied in real-time to adapt the clusters and the cluster set to short-term behavioral changes.

5 Performance Evaluation

We conducted several experiments to: 1) compare the efficacy of the path features in characterizing user sessions, 2) study the accuracy of our similarity measure in detecting the similarity among user sessions, 3) compare the performance of the FM model with that of the traditional Vector model, and 4) investigate the accuracy of the dynamic clustering. Here, we summarize the results of these experiments. The detailed description of the results, and also our experimental methodology is included in [11].

5.1 Efficacy of the Path Features

A set of experiments was conducted to study the relative efficacy of the path features H and S in detecting similarities between user sessions. In Equation 1, the weight factor w_i indicates relative importance of the path feature F_i in computing the aggregated similarity measure. The higher weights are assigned to the features that are more effective in capturing the similarities. Our experiments were intended to find the compromised set of weight factors w_S (weight factor for S) and w_H that results in the optimum accuracy in capturing the similarities.

The experiment results show that regardless of the weight factors applied, the accuracy is always above 94%. Thus, both features (Hit and Sequence) are equally successful in identifying the spatial similarities. Depending on distinguishability of the dataset, the optimum accuracy is achieved by employing a compromised combination of the similarities detected in *Hit* and *Sequence*. In brief, when similarity among users of the same cluster decreases, it is more important to track which pages they visit (*Hit*) rather than where in the session they visit each page (*Sequence*).

5.2 Accuracy of the Similarity Measures

In Section 3, we introduced *PPED* as an accurate similarity measure. Here, we compare accuracy of *PPED* with two classical similarity measures, i.e. *PED* and cosine (*VA*).

The experiment results demonstrate that for real data, which assumes low distinguishability, *PPED* outperforms *VA* with a wide margin. The results also show that since *PPED* can measure the similarity between a user and a cluster based on user characteristics rather than cluster characteristics, overestimation

of the distance between the session and its intended cluster is avoided by disregarding unnecessary cluster characteristics in distance estimation. Therefore, *PPED* can achieve up to 30% improvement in accuracy as compared to *PED*.

5.3 Performance of the FM Model

We conducted some experiments to compare performances of a sample FM model, namely $FM^{(2)}$ with H and S as its features, with the traditional Vector model, which is considered equivalent to $FM^{(1)}$ with H as its only feature.

Results of this study demonstrate that accuracy of the Vector model decreases as the user sessions become less distinguishable, while the FM model can reasonably maintain its accuracy even with highly indistinguishable datasets. This superiority is because of: 1) incorporating *Sequence* into the model, and 2) capturing features based on order-2 segments.

5.4 Performance of the Dynamic Clustering

In Section 4, we introduced *dynamic clustering* as an approach to update cluster models in real-time. However, we also mentioned that dynamic clustering trades accuracy for adaptability. We conducted several experiments to study the degradation of the accuracy due to applying the dynamic clustering. For this purpose, we compared dynamic clustering with K-Means.

The experiment results show that as expected accuracy of dynamic clustering is less than accuracy of K-Means but the degradation of the accuracy is tolerable. Thus, the dynamic clustering can be applied to achieve adaptability but it should be complemented by long-term periodical re-clustering. The results also demonstrate that the performance of the dynamic clustering is much better in updating the existing clusters as compared to creating new clusters.

6 Conclusions

We defined a new model, the FM model, which is a generalization of the Vector model to allow for flexible, adaptive, and real-time Web Usage Mining (WUM). We argued that these characteristics are not only useful for off-line and conventional WUM, but also critical for on-line anonymous WUM. We demonstrated how flexibility of FM allows conceptualization of new navigation features as well as trading performance for accuracy by varying the *order*. For FM, we proposed a similarity measure, *PPED*, that can accurately classify partial sessions. This property is essential for real-time and anonymous WUM. We then utilized *PPED* within a dynamic clustering algorithm to make FM adaptable to short-term changes in user behaviors. Dynamic clustering is possible since unlike the Markov model, incremental updating of the FM model has a low complexity. Finally, we conducted several experiments, which demonstrated the high accuracy of *PPED* (above 98%), the superiority of FM over the Vector model (by at least 25%) and the tolerable accuracy of dynamic clustering as compared to K-Means (only 10% worse), while being adaptable.

Acknowledgments. This research has been funded in part by NSF grants EEC-9529152 (IMSC ERC) and ITR-0082826, NASA/JPL contract nr. 961518, DARPA and USAF under agreement nr. F30602-99-1-0524, and unrestricted cash/equipment gifts from NCR, IBM, Intel and SUN.

References

1. Cadez I., Heckerman D., Meek C., Smyth P., and White S.: Visualization of Navigation Patterns on Web-Site Using Model Based Clustering. Technical Report MSR-TR-00-18, Microsoft Research, Microsoft Corporation, Redmond, WA, (2000)
2. Huberman B., Pirolli P., Pitkow J., and Lukos R.: Strong Regularities in World Wide Web Surfing. *Science*, 280, p.p 95-97 (1997)
3. Mobasher B., Cooley R., and Srivastava J.: Automatic Personalization Based on Web Usage Mining. Special Section of the Communications of ACM on "Personalization Technologies with Data Mining", 43(8):142-151 (2000)
4. Mulvenna M.D., Anand S.S., Büchner A.G.: Personalization on the Net using Web mining: Introduction. *CACM* 43(8): 122-125 (2000)
5. Perkowitz M., Etzioni O.: Toward Adaptive Web-Sites: Conceptual Framework and Case Study. *Artificial Intelligence* 118, p.p 245-275 (2000)
6. Shahabi C., Zarkesh A.M., Adibi J., and Shah V.: Knowledge Discovery from Users Web Page Navigation. *IEEE RIDE97 Workshop*, April (1997)
7. Spiliopoulou M.: Web usage mining for site evaluation: Making a site better fit its users. Special Section of the Communications of ACM on "Personalization Technologies with Data Mining", 43(8):127-134 (2000)
8. Srivastava J., Cooley r., Deshpande M., Tan M.N.: Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations*, Vol. 1, Issue 2 (2000)
9. Yan T.W., Jacobsen M., Garcia-Molina H., Dayal U.: From User Access Patterns to Dynamic Hypertext Linking. Fifth International World Wide Web Conference, Paris, France, (1996)
10. Zarkesh A., Adibi J., Shahabi C., Sadri R., and Shah V.: Analysis and Design of Server Informative WWW-Sites. *ACM CIKM '97* (1997)
11. Shahabi C., Banaei-Kashani F., Faruque J., Faisal A.: Feature Matrices: A Model for Efficient and Anonymous Mining of Web Navigations. Technical Report USC-CS-00-736, Computer Science Department, University of Southern California.

Hypertext References

HREF1: <http://www.personify.com>

HREF2: <http://www.websidestory.com>

HREF3: <http://www.webtrends.com>

Faceted Preference Matching in Recommender Systems

Fred N. Loney

Spirited Software, Inc.

floney@spiritedsw.com

Abstract. A recommender system assists customers in product selection by matching client preferences to suitable items. This paper describes a preference matching technique for products categorized by a faceted feature classification scheme. Individual ratings of features and products are used to identify a customer's predictive neighborhood. A recommendation is obtained by an inferred ranking of candidate products drawn from the neighborhood. The technique addresses the problem of sparse customer activity databases characteristic of e-commerce. Product search is conducted in a controlled, effective manner based on customer similarity. The inference mechanism evaluates the probability that a candidate product satisfies a customer query. The inference algorithm is presented and illustrated by a practical example.

1 Introduction

E-commerce sites can attract and retain customers by fostering a virtual community of users sharing a common interest. A community focus has two complementary aspects:

- collaboration in integrating member-generated content
- personalization based on client preference

A recommender system enables a community focus by matching personal preferences to shared product evaluations. The collaborative ratings are used as a source of evidence to extrapolate a user's observations to an unfamiliar item.

Recommender systems exhibit the following characteristics:

- active participation – users contribute product ratings and feature preferences
- sparse coverage – the number of products purchased or rated by an individual is a small proportion of the total number of available products
- probabilistic inference – the recommendation is understood to be an approximate match based on the available information
- selectivity bias – it is more important to avoid a false positive than a false negative

User contributions take a variety of forms. The least informative contribution is a record of purchase activity. Suggestions are based on prior purchases, on the assumption that a purchase is an implicit recommendation. Confidence is increased if the user provides feedback by rating the purchased product. Capturing user preferences of product features offers additional useful information.

Sparse databases present a special challenge for building a recommender system. The critical function of a recommender system is to find a *predictive neighborhood* of users who have enough in common to make a reasonably accurate prediction of

shared likes and dislikes. A recommendation is essentially an inference based on uncertain supporting evidence. However, the selectivity bias implies that some errors are worse than others. Overlooking a good product is less egregious than recommending a bad product.

This paper presents a technique for making recommendations for active systems that takes into account both product ratings and feature preferences. Particular attention is paid to finding a suitable neighborhood and factoring features into the inference process. The paper is organized as follows: Section 2 is a cursory review of related work. Section 3 presents an algorithm for identifying a predictive neighborhood. Section 4 incorporates feature preferences into the technique. The final section summarizes the results and discusses future work.

2 Related Work

Recommender systems have been implemented for such diverse applications as office systems [4], news groups [12], music [17] and movies [8]. A comprehensive review of e-commerce recommender systems is presented in [15].

Some algorithms for identifying comparable individuals include the LikeMinds closeness function [16], the constrained Pearson coefficient [17], the Spearman rank correlation [7], "cosine" vector similarity [2] and the Intelligent Recommendation Algorithm (IRA) [1]. The constrained Pearson coefficient is chosen as the basis for the technique described here because it offers good results in most situations [7] and is easily adapted to the normalized form presented in Section 3. The correlation is derived from the ratings of two users a and b over a common set of products P . For recommendations r_{ap} and r_{bp} of a product $p \in P$, the general form of the constrained Pearson coefficient is given by:

$$corr_{ab} = \frac{\sum_{p \in P} (r_{ap} - \bar{r}_a)(r_{bp} - \bar{r}_b)}{\sqrt{\sum_{p \in P} (r_{ap} - \bar{r}_a)^2 (r_{bp} - \bar{r}_b)^2}} \quad (1)$$

where \bar{r}_a is the mean of a 's observations and \bar{r}_b is the mean of b 's observations.

These algorithms treat an item as an opaque entity without elaborating item features. The recommendation encapsulates the item as a package of features without considering the contribution of any particular feature to that item's appeal. Facets [11] offer a way to categorize features and a framework to rank and compare features as well as products. Facets have been used to index and retrieve relevant software reuse components [11,3]. They are adapted to recommender systems in this paper as a feature classification mechanism.

3 Neighborhood Formation

A neighborhood is formed by discovering users who have similar ratings. This section presents a technique to identify a neighborhood based on standardized product ratings. Section 4 extends this basic model to incorporate feature preferences into the user comparisons as well.

3.1 Normalization

Typically, a rating is an integer in the range $[1, v]$, where v is a small positive integer. We can, without loss of generality, normalize the set of ratings to the unit interval $[-1, 1]$ with zero mean. Normalization standardizes and simplifies subsequent rating comparisons and recommendation inference. For a set of ratings R where $1 \leq r \leq v, r \in R$, the normalization transformation is defined by:

$$(r) = \frac{2r - v - 1}{v - 1} \quad (2)$$

This is a well-formed linear transformation that maps the corresponding bounds and mean, since:

$$\begin{aligned} (1) &= \frac{2 - v - 1}{v - 1} = \frac{v + 1}{v - 1} = -1 \\ (v) &= \frac{2v - v - 1}{v - 1} = 1 \\ \left(\frac{v+1}{2}\right) &= \frac{(v+1) - v - 1}{v - 1} = 0 \end{aligned} \quad (3)$$

Furthermore, is strongly order-preserving, i.e. for $r_1, r_2, r_3 \in R$

$$\begin{aligned} r_1 < r_2 &\quad (r_1) < (r_2) \\ r_1 = r_2 &\quad (r_1) = (r_2) \\ r_1 > r_2 &\quad (r_1) > (r_2) \end{aligned} \quad (4)$$

and

$$\frac{r_1 - r_2}{r_1 - r_3} = \frac{(r_1) - (r_2)}{(r_1) - (r_3)}$$

[10] presents the generalized normalization procedure and proof of correctness. Henceforth, it is assumed that all rating sets are so normalized.

3.2 Similarity

The normalized Pearson correlation $corr_{ab}$ is given by Equation (5):

$$corr_{ab} = \frac{\sum_i r_{ai} r_{bi}}{\sqrt{\sum_i r_{ai}^2} \sqrt{\sum_i r_{bi}^2}} \quad (5)$$

The Pearson correlation has the desirable property that a user whose ratings are consistently inflated or deflated by a positive constant multiplier with respect to a base user preserves a high correlation with the base user, i.e. if $r_{bi} = c \cdot r_{ai}$ and $c > 0$ then

$$corr_{ab} = \frac{\sum_i r_{ai} \cdot c r_{ai}}{\sqrt{\sum_i r_{ai}^2} \sqrt{\sum_i (c r_{ai})^2}} = \frac{c \sum_i r_{ai}^2}{|c| \sqrt{\sum_i r_{ai}^2} \sqrt{\sum_i r_{ai}^2}} = 1 \quad (6)$$

On the other hand, Pearson correlation lacks a useful feature of the IRA algorithm for detecting inverse correlations, whereby the multiplier c approaches -1 . A user with a strongly negative rating correlation is conferred a high positive predictive value. The IRA algorithm encodes this negative correlation inversion such that the similarity metric approaches 1 as the correlation approaches -1 . In fact, a negative correlation carries the same weight as a inversely positive correlation.

The motivation is to encode the predictive value of consistently different individuals. While a negative correlation may be a useful predictor, it is unclear that it applies in all domains. Furthermore, it is unlikely that a negative correlation should carry the same weight as a comparably positive correlation. Intuitively, most customers would prefer a recommendation of an item highly rated by consistently like-minded individuals to a recommendation of an item poorly rated by consistently different individuals.

Given the similarity bias mentioned in the introduction, positive recommendations of poorly rated items should be exercised with discretion. The approach taken here is to allow for this and other judgements of user credibility as an explicit scaling factor. A user b is assigned a *credibility* with respect to user a that signifies the credence placed in user b 's judgement. The credibility, denoted $cred_{ab}$, is a scaling factor of the correlation in the range $[-1, 1]$. It is typically calculated based on simple heuristics, user feedback or self-assessments. For example, a professional wine critic might be accorded higher credibility than other evaluators of wine products.

The conventional Pearson correlation assumes $cred_{ab} = 1$ for all users a, b . The IRA algorithm assumes the two-valued credibility assignment:

$$cred_{ab} = \begin{cases} 1, & corr_{ab} \geq 0 \\ 1, & corr_{ab} < 0 \end{cases}$$

Other heuristics are possible, and can be discovered and adjusted with experience in the application domain.

Given the correlation and credibility, the *confidence* imputed to user a in the ratings of user b is given by:

$$conf_{ab} = cred_{ab} \cdot corr_{ab} \quad (7)$$

Finally, a *predictor neighborhood* of user a with confidence K is the set of users

$$N_a = \{u \mid U \mid conf_{ab} > K\} \quad (8)$$

The value of K is chosen to balance predictive support with an adequate neighborhood size. In practice, it is convenient to set a standard threshold value of K subject to a minimum neighborhood size. The experimental evidence in [13] is useful to gauge an appropriate neighborhood size based on the size and dimensionality of the observation set.

4 Facets

Establishing a feature set with a predictive capacity similar to observer coherence lends additional credence to a recommendation. Features serve three purposes in recommender systems: neighborhood formation, query specification and recommendation inference. After introducing a facet classification scheme, this section describes how to use feature correlations to form neighborhoods and feature templates to format queries.

4.1 Faceted Classification

A useful technique for assigning features to items is to partition the features into pre-defined categories, or *facets*. A simple faceted classification of two wines is presented in [Table 1](#).

Table 1. Faceted Wine Classification

<u>Winery</u>	<u>Vintage</u>	<u>Grape</u>	<u>Flavor</u>
Arbor Springs	1999	Chardonnay	oak, vanilla
Chloe Vineyards	1996	Pinot Noir	black pepper

There are no a priori restrictions on feature cardinality besides those imposed by the application domain. A wine may be required to have one winery, but may have any number of flavor descriptors. Furthermore, a feature can take a range of values indicating the degree to which a product manifests the feature, as in Table 2.

Table 2. Feature Valuation

<u>Winery</u>	<u>Vintage</u>	<u>Grape</u>	<u>Flavor</u>
Arbor Springs	1999	Chardonnay	oak=0.8 vanilla=0.7

Additional features may be derived from assigned features. For example, the Winery facet determines a Region that is common to all wines from that winery. Similarly, Grape can be grouped into the Color features *white* or *red*¹. Thus, a facet is either *assigned* if its features are directly dependent on the product, or *derived* if its features are dependent, directly or indirectly, on an assigned facet.

The hierarchical classification induces a subsumption ordering: facet F_i *subsumes* facet F_j if the value of F_j determines the value of F_i . This ordering is reflexive, transitive and antisymmetric. There is a natural extension of the facet subsumption ordering to tuples:

$$F_i \text{ subsumes } F_j \text{ if and only if } F_i \text{ subsumes } F_j, 1 \leq i \leq m.$$

The subsumption ordering induces a lattice, as in Figure 1.

¹ Even this simple grouping hides subtle data modeling issues. The common assortment of wines into the three colors red, white and rosé invalidates the Grape ? Color dependency, since rosé is a wine-making style that is a feature of the particular wine rather than the grape. The potential for ambiguity and anomaly motivate a predefined stable facet classification by a domain expert.

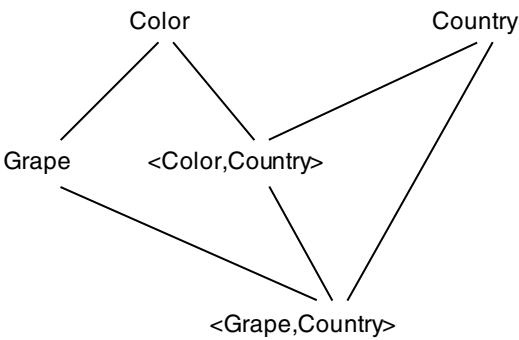


Fig. 1. Facet tuple space for wine grape and origin

A value in facet tuple space constitutes a *feature template* representing a unique combination of features. A customer expresses the desirability of a feature or combination of features by assigning a rating to a feature template. Table 3 shows ratings of feature templates taken from the facet tuple spaces Region Grape, Country Color and Region.

Table 3. Feature rating

User	Feature Template	Rating
Mel	<Oregon,Pinot Noir>	0.8
Mel	<Germany,White>	0.7
Maurice	<Mosel>	0.9

4.2 Feature Correlation

Facets facilitate neighborhood formation by permitting additional opportunities for discovering predictors. The user correlation defined in Equation (6) is readily adapted to compare feature preferences. However, a feature rating must first be tempered by the generality of the feature, since a generic feature has less predictive value than a specific product in determining customer similarity. For example, a match on ratings of a particular wine wine has higher predictive value than a match on the feature template <Oregon,PinotNoir>, which in turn has higher predictive value than a match on the feature <Red>.

The predictive value is captured by the *coherence* of a feature template, $coh(\langle f_1,...,f_n \rangle)$, where $0 \leq coh(\langle f_1,...,f_n \rangle) \leq 1$. Coherence can be assigned by a domain expert or inferred by a heuristic. There are three types of heuristics:

- an *aggregation* heuristic determines the coherence of a composite feature template as the joint product

$$coh(\langle f_1,...,f_n \rangle) = \prod_{i=1}^n coh(f_i)$$

(9)

- a *classification* heuristic applies a default facet coherence, e.g. $\text{coh}(\text{Germany}) = \text{coh}(\text{Country})$.
- an *inheritance* heuristic uses the value of a subsumption facet, e.g. $\text{coh}(\text{PinotNoir}) = \text{coh}(\text{Red})$.

The adjusted feature rating is then given by

$$\hat{r}_{af} = r_{af} \cdot \text{coh}(f_i)(r_{af} - \bar{r}) \quad (10)$$

This has the effect of attenuating the strength of the feature rating by the coherence. The feature rating is thus weaker than a product rating to the extent that the feature coherence is weaker. The adjusted features ratings are then included in the overall correlation that determines user similarity.

4.3 Query Specification

A query specification expresses user preference for features in the recommendation. The feature ratings described in Section 3.2 act as a default query specification, since they capture a user's customary preferences. Alternatively, an ad hoc query can be formulated explicitly by assigning preference values to feature templates.

Query templates can be combined using the logical “and”, “or” and “not” operators to form a logical expression. The general form of a query Q with n terms, then, is $Q = \{q_i = v_i, i = 1, \dots, n\}$ where each condition q_i is a feature template or a logical expression whose operands are query subconditions.

The value v assigned to a condition measures the desirability of the feature combination in the query result. $v = 1$ if a candidate product is required to match the condition. $v = -1$ if the feature combination is disallowed in the result. $v = 0$ if the requestor is indifferent about the occurrence of the feature combination. The query specification of Equation (11) below indicates a slight preference for Oregon red wine and a strong preference to show Pinot Noir in the result.

$$Q = \langle \text{Oregon, Red} \rangle = 0.2, \langle \text{PinotNoir} \rangle = 0.8 \quad (11)$$

Given a query condition $q = v$, define an adjusted value $\hat{v}(q) = v \cdot (1 - \text{freq}(q))$, where $\text{freq}(q) = \frac{\text{card}(\{p \mid p \text{ matches } q\})}{\text{card}(P)}$ is the frequency of occurrence of products satisfying q . Similarly, the adjusted value for a product that does not satisfy the condition is $\hat{v}(q) = -v \cdot \text{freq}(q)$. The probability that a product will be preferred over another product is then given by:

$$\text{prob}(q \mid p) = \begin{cases} \frac{\hat{v}(q) + 1}{2} & \text{if } p \text{ matches } q \\ \frac{\hat{v}(q) + 1}{2} & \text{otherwise} \end{cases} \quad (12)$$

For the query template $Q_n = \{q_i = v_i, i = 1, \dots, n\}$, the joint adjusted value is defined recursively as:

$$\hat{v}(Q_1) = \hat{v}(q_1) \quad (13)$$

$$\hat{v}(Q_k) = \hat{v}(q_n) + \hat{v}(Q_{n-1}) - \hat{v}(q_n) \cdot \hat{v}(Q_{n-1})$$

Table 4 shows an example valuation for the query specification in Equation (11). The joint valuation for a product satisfying both query conditions in the example is $\hat{v}(Q) = .19 + .72 - (.19)(.72) = 0.7737$ and the joint probability is $prob(Q) = 0.88685$.

Table 4. Example query distribution and valuation

q	$freq(q)$	v	$\hat{v}(q)$	$prob(q)$	$\hat{v}(q)$	$prob(q)$
<Oregon,Red>	.05	.20	.19	0.595	-.01	0.495
<PinotNoir>	.10	.80	.72	0.86	-.08	0.46

Query condition valuations are assumed to be independent, e.g. the preference expressed for an Oregon red wine in the subcondition <Oregon,Red> of query Q in Equation (11) above is 0.2, regardless of whether the wine is a Pinot Noir. Cross-dependencies can be factored into the query by use of logical operators, e.g.

$$\langle \text{Oregon,Red} \rangle \text{ and not } \langle \text{PinotNoir} \rangle = 0.2$$

4.4 Recommendation Inference

The task of recommendation inference is to use the neighborhood and feature ratings to assess the probability that a product will match a user query. This is evaluated by calculating the probability that each candidate product would be preferred over another product. The neighborhood is restricted to users who rate the product. The probability $prob(Q|p)$ that user u would prefer product p maps the user rating to the interval $[0, 1]$:

$$prob(u|p) = \frac{r_{up} + 1}{2} \quad (14)$$

$$prob(u|p) = 1 - prob(u|p)$$

The probability of a neighborhood recommendation is the average confidence in the users rating the product, expressed as a probability in the range $[0, 1]$:

$$prob(N|p) = \frac{card(N) + conf_{au} r_{up}}{2 \cdot card(N)} \quad (15)$$

The aggregate probability that the product p is the preferred product to satisfy query condition q is the weighted cumulative probabilities of the neighborhood recommendation and the facet recommendation:

$$prob(q) = prob(N|p) + (1 - prob(N|p)) \cdot prob(Q|p) \quad (16)$$

The weight is a value in the range $[0, 1]$ that expresses the relative contribution of the neighborhood assessment *vis-a-vis* facet template matching.

An example inference is presented graphically in Figure 2. The nodes represent predictive sources, annotated with the probability that the subject product p is preferred over other products at that node. The query specification Q is given by **Table 4** and the neighborhood N consists of two users u_1 and u_2 who rate p 0.6 and 0.8, resp., with confidence value 1 for each user. The probability assigned to a user node is given by Equation (14) as $prob(u \mid p) = (r_{up} + 1)/2$, or $p=0.8$ for u_1 and $p=0.9$ for u_2 . The joint probability for neighborhood N is given by Equation (15) :

$$prob(N \mid p) = \frac{card(N) + \frac{conf_{au} r_{up}}{u \ N}}{2 \ card(N)} = \frac{2 + 0.6 + 0.8}{4} = 0.85 \quad (17)$$

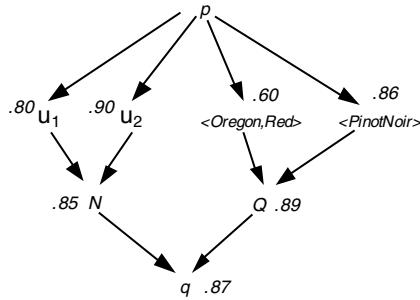


Fig. 2. Example query evaluation

The contribution of the product feature templates is given in Equations (12) and (13). The final probability for q is given by Equation (16) with a nominal weight of 0.5:

$$prob(q \mid p) = 0.5 \ prob(N \mid p) + 0.5 \ prob(Q \mid p) = (.5 \ .85) + (.5 \ .89) = 0.87 \quad (18)$$

This is the probability that p will be preferred over another wine, given the available evidence. Each candidate product would then be evaluated and ranked according to resulting probability.

5 Conclusion

In this paper, a technique was presented for recommending products based on product and feature ratings. The technique is appropriate for an active recommender system, which entails the application of domain expertise in devising a suitable facet classification scheme. The execution is intentionally dependent on domain-specific judgements and is sensitive to the values chosen. Faceted recommendation is useful to the extent that the application domain can be structured by accepted, well-understood facets.

Future work is required to assist the domain expert in building the inference model in two respects: *i*) sensitivity analysis of the value assignments and *ii*) using observation sets to learn the relative contribution of features to user preference. A fruitful avenue of investigation is to represent the model as a Bayesian network [6,13] and apply techniques of sensitivity analysis [9] and learning [5].

References

1. Aggarwal, C., Wolf, J., Wu, K. and Yu, P. Horting Hatches an Egg: a New Graph-theoretic Approach to Collaborative Filtering. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA (1999) 201-212
2. Breese, J., Heckerman, D. and Kadie, C. Empirical analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence* (1998) 43-52
3. Damiani, E., Fugini, E. and Bellettini, C. A Hierarchy-Aware Approach to Faceted Classification of Object-Oriented Components. *ACM Transactions on Software Engineering and Methodology* 8:3 (1999) 215-262
4. Goldberg, D., Nichols, D., Oki, B. and Terry, D. Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM* 35:12 (1992) 61-70
5. Heckerman, D. *A Tutorial on Learning with Bayesian Networks*. Microsoft Technical Report MSR-TR 95-06 (1995)
6. Heckerman, D. and Wellman, M. Bayesian Networks. *Communications of the ACM* 38:3 (1995) 27-30
7. Herlocker, J., Konstan, J., Borchers, A. and Riedl, J. An Algorithmic Framework for Performing Collaborative Filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA (1999) 230-237
8. Hill, W., Stead, L., Rosenstein, M. and Furnas, G. Recommending and evaluating choices in a virtual community of use. In *Proceedings of ACM Conference on Human Factors in Computing Systems*, Denver, CO (1995) pages 194-201.
9. Howard R. and Matheson, J. (eds.). *The Principles and Applications of Decision Analysis*. Strategic Decisions Group, Menlo Park, CA (1983)
10. Loney, F. Normalization of a Bounded Observation Set (or How to Compare Apples and Oranges). Spirited Software Technical Report SSI-TR 2001-01, available at www.spiritedsw.com/pub/techreports/tr2001-01.pdf (2001)
11. Prieto-Díaz, R. Implementing faceted classification for software reuse. *Communications of the ACM* 34:5 (1991) 88-97
12. Resnick, P., Iacovou, N. Suchak, M., Bergstrom, P. and Riedl, J. GroupLens: an Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of the Fifth ACM Conference on Computer Supported Cooperative Work*, Chapel Hill, NC (1994) 175-186
13. Ribeiro, B. and Muntz, R. A Belief Network Model for IR. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Zurich, Switzerland (1996) 253-260
14. Sarwar, B., Karypis, G., Knowtan, J. and Riedl, J. Analysis of Recommendation Algorithms for E-Commerce. In *Proceedings of the Second ACM Conference on Electronic Commerce*, Minneapolis, MI (2000) 158-167
15. Schafer, B., Konstan, Riedl, J. Recommender Systems in E-Commerce. In *Proceedings of the First ACM Conference on Electronic Commerce*, Denver, CO (1999) 158-166
16. Schapire, R. and Singer, Y. Improved boosting algorithms using confidence-rated predictions. *Machine Learning* 37:3 (1999) 297-336
17. Shardanand, U. and Patti Maes, P. 1995. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of ACM Conference on Human Factors in Computing Systems*, Denver, CO (1995) 210-217

Pinpoint Web Searching and User Modeling on the Collaborative Kodama Agents

Tarek Helmy, Satoshi Amamiya, and Makoto Amamiya

Department of Intelligent Systems
Graduate School of Information Science and Electrical Engineering

Kyushu University
6-1 Kasugakoen, Kasuga-shi
Fukuoka 816-8580, Japan

E-mail: [helmy,roger,amamiya]@al.is.kyushu-u.ac.jp
Tel: 81-92-583-7615 & Fax: 81-92-583-1338

Abstract. The primary application domain of Kodama¹ is the WWW and its purpose in this application is to assist users to find desired information. Three different categories of Kodama's agents are introduced here, Web Page Agents, Server Agents, and User Interface Agents. Kodama agents learn and adapt to the User's Preferences (UP), which may change over time. At the same time, they explore these preferences to get any relevancy with the future queries. These communities of Kodama agents autonomously achieve and update their Interpretation Policies (IP) & UP and cooperate with other agents to retrieve distributed relevant information on the Web. This paper studies ways to model user's interests and shown how these models can be deployed for more effective information retrieval. In terms of adaptation speed, the proposed methods make Kodama system acts as a pinpoint information retrieval system, converges to the user's interests and adapts to the user's sudden change of interests.

Keywords: Web Data Mining and Analysis, Collaborative Information agents, Web site Management, Adapting to User's Model.

1 Introduction

With the exponentially growing amount of information available on the Internet, the task of retrieving relevant information consistent with the user's information need has become increasingly difficult. The model behind Traditional Search Engines (TSE) analyzes collected documents once to produce indices, and then amortizes the cost of such processing over a large number of queries, which access the same index. This model assumes that the environment is static. The Web is however highly dynamic, with new documents being added, deleted, changed, and moved all the time, rapid growth rate and dynamic nature [13, 14]. So, at any given time an index of the TSE will be somewhat inaccurate and somewhat incomplete. Also users normally face with very large hit lists with low precision. Moreover, the information gathering and retrieving processes in TSE are independent of user's preference, and therefore feedback from the later process is hardly adaptive to improve the quality of the former process. Kodama project starts in response to the need for a new kind of search engine that completely different from the frustrating, time-consuming search engines that populated the Internet. Kodama set out to create a new type of search experience that provides the user with a relevant information. Researchers in Artificial Intelligence

¹ Kyushu University Open Distributed Autonomous Multi-Agent

(AI) and IR fields have already succeeded in developing agent-based techniques to automate tedious tasks and to facilitate the management of information flooding [3, 4, 15]. A way to partially address the scalability problems posted by the size and dynamic nature of the Web is to decentralize the index-building process. Dividing the task into localized SAs that agentify specific domains by a set of WPAs developed in this project [6]. The success of this project has been achieved by the cooperation among the WPAs [5, 6, 7]. Kodama is a distributed multi-agent for the IR in large, dynamic and distributed environment such as WWW. In this paper we will start by describing the mechanism of agentifying a Web site and creating WPAs communities. We then discuss our new methodologies of calculating the relevancy of retrieved Web page contents to the UP, which is used in UIA and WPA. We discuss how UIA in Kodama system makes use of user's query history and bookmark files as the UP. We also describe the autonomy techniques used in WPA and UIA. Finally we present the experimental results and future work of Kodama system.

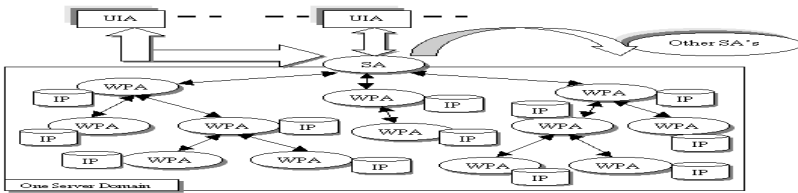


Fig. 1. The hierarchical structure of an agentified domain

2 Web Site Agentification

Cooperating intelligent Kodama agents are employed to agentify the Web where the hyper structure is preexisting in the form of Web links [12, 16]. Our system uses three types of Kodama agents in the agentification (Fig. 1) for searching the Web. A Server Agent (SA) assigned to each Web server, a Web Page Agent (WPA) assigned to each Web page, and a User Interface Agent (UIA) assigned to each user. There is a clear mapping between the problem of searching the Web in Kodama system and the classic AI search paradigm. Each WPA of the agentified Web pages is a node, and the hypertext links to the other down chain WPAs are the edges of the graph to be searched. In typical AI domains a good heuristic will rate nodes higher as we progress towards some goal node. In the Kodama system domain, the heuristic models how a page is relevant to the given query. A standard best-first search algorithm is used by the SA and the WPAs while interacting with the down chain agents. It has been slightly modified so that it will finish after reaching a predefined depth value, and return the best WPAs, which have a good relevancy to the given query.

2.1 Web Page Agent

The software designer has been responsible for providing each agent with its Interpretation Policy [IP] so far [9]. In Kodama this is done automatically by the WPA itself. At the initialization phase, each WPA analyzes the content of its Web page starting with the base address given by the SA with a serial ID number. Each WPA has its own parser, to which the WPA passes a URL, and a private Interpretation Policy (IP), in which the WPA keeps all the policy keywords, found in its URL. The WPA takes essential properties and principles given by the SA to produce the IP as follows. The WPA sets the URL of that Web page as its name, loads the HTML document of that Web page, parses the HTML document, eliminates the noisy words, and stemming

a plural noun to its single form and inflexed verb to its infinitive form. After that, the WPA creates its IP using an additional heuristics, in which additional weights are given to words in the title and headings of the Web page. An IP is used by the WPA to decide whether or not the keywords in the query belong to the WPA. At the retrieval phase, WPAs, when received a user's query from the SA initiate search by interpreting the query and/or either asking '*Is this yours?*' or announcing '*This is yours,*' to its down-chain WPAs. The selected WPAs and/or their down-chain WPAs of each Web server, in turn, interpret the query according to their IPs and reply the answer '*This is mine*' or '*Not mine*' with some confidence.

2.2 Server Agent and WPAs Community

A SA is assigned to one Web server to be responsible. The SA starts from the portal address of the Web server and creates the hyper structure of WPAs communities based on the hyper link structure in the Web server. The SA knows all WPAs in the server and works as a gateway when WPAs communicate with each other or with one in another server. The SA initiates all WPAs in its server when it starts searching relevant information to the user's query. The SA clusters the WPAs into communities and automatically defines its attributes, to be used in the routing mechanism as we mention in the future work. We introduce a definition of WPAs community that enables the SA to effectively focus on narrow but topically related subset of WPAs and to increase the precision of searching results. We define a community to be a set of WPAs has a common directory name. The SA uses URL's implied information (e.g. the physical directory name as key words) for clustering the WPAs into communities. The SA applies filtration and weighting mechanisms over these directory names to define precisely these attributes.

2.3 User Interface Agent

The UIA is implemented as a browser independent Java application. Monitoring the user-browsing behavior is accomplished via a proxy server that allows the UIA to inspect HTTP requests from its browser. The UIA resides in the user's machine, communicates with the WPAs via an SA to retrieve information relevant to the user's query, and shows the results returned by the WPAs to the user after filtering and re-ranking them. It receives user's responses of his/her interest/not interest to the results and regards them as rewards to the results. In contrast to other systems that learn a UP and use it to determine relevant documents [14, 15], UPs of Kodama's users are continuously evolving according to the dynamically changing of UPs. The UIAs in Kodama system look over the shoulders of the users and record every action into the query history file. After enough data has been accumulated, the system uses this data to predict a user's action based on the similarity of the current query to already encountered data. The followings are the job stream of the UIA.

- (1) The user starts by sending a Natural Language (NL) query to the UIA.
- (2) UIA analyzes the NL query using a simple NL processing algorithm, throws out irrelevant words and transforms it to Q_{in} .
- (3) The UIA calculates the similarity with the method described here and looks for relevant URLs in UP files using equations 5, 6.
- (4) If UIA finds relevant URLs in UP then shows them and asks the user whether the user is satisfied or wants to search the Web.
- (5) In case of finding relevant queries in UP, the UIA takes two queries from the UP, whose similarity to the given query is over a predefined threshold value and

concatenates the given query with the keywords of these two queries after removing the redundant terms to expand Q_{in} .

- (6) If UIA could not find in its UP files any URLs relevant to Q_{in} then UIA routes Q_{in} to a relevant SA, which in turn forwards it to its community of WPAs.
- (7) The UIA receives the search results returned by the WPAs via the SA. The results consist of a set of contents of Web pages.
- (8) The UIA takes a set of queries, whose similarity to Q_{in} is over a predefined threshold value from the UP to expand Q_{in} . Then, the UIA makes a context query from them and Q_{in} , to be used for filtering the retrieved documents.
- (9) The user either explicitly marks the relevant documents using UIA's feedback or the system implicitly detects user's response.

In the current version, the relevant SA to the user's query selected either by having the user explicitly define the SA as a specific portal or by having the system determine by itself by examining the user's hot-list, query history and SA's attributes.

2.4 Relevancy with User's Query History and Bookmark by UIA

Recording and analyzing user's accessing histories and bookmark by the UIA are quite important to catch his/her preferences. The query history file contains information about previously visited URLs for specific queries, the number of occurrences that this URL is visited, the time of visiting & leaving and the query. The bookmark file contains a user's hot-list of Web links, the number of occurrences that a URL is visited, bookmarking time of the URL and its title. The query and the title fields in query history and bookmark files are represented as a vector of keywords sorted in alphabetical order, where a weight value is assigned to each keyword to reflect the correlation with the content of the page. User's responses (), for instances, are *Useless*, *Not very useful*, *Mildly interesting*, *Neutral*, *Interesting* or *Bookmark* and each has a value between 0 and 1. When looking up relevant URL from the UP, the UIA calculates similarities as follows:

First: We define equations to calculate the similarity between a user's query and his/her query history file. Assume we have a query history file and a bookmark file of n URL lines gathered. $Q_{in} = \langle k_1, k_2, \dots, k_n \rangle$ stands for a vector of keywords sorted in alphabetical order, of the query given by the user. $Q_j = \langle K_{j,1}^h, K_{j,2}^h, \dots, K_{j,m}^h \rangle$, ($1 \leq j \leq n$) stands for the vector sorted in alphabetical order, of the query of j th line in the user's query history file, where $K_{j,i}^h = k_{j,i}^h w_{j,i}^h$, $k_{j,i}^h$ is the i th keyword in the j th line and $0 \leq w_{j,i}^h \leq 1$ is its weight. Similarly, $T_j = \langle K_{j,1}^b, K_{j,2}^b, \dots, K_{j,l}^b \rangle$ and $K_{j,i}^b = k_{j,i}^b w_{j,i}^b$ are defined for the title of j th line in the user's bookmark file. The weight $w_{j,i}^h$ and $w_{j,i}^b$ are incrementally computed with the number t_j of visiting to URL_j .

$$w_{j,i}(t_j + 1) = w_{j,i}(t_j) + (1 - w_{j,i}(t_j)) \quad (1)$$

Where $w_{j,i}$ means $w_{j,i}^h$ or $w_{j,i}^b$, and $0 \leq 1$ is a user's response described above.

Initial value $w_{j,i}(1)$ is set by the user's first response. $0 < 1$ is a function of t_j , i.e., (t_j) , and (t_j) depends on how long user's response history upon the keyword will be involved in calculating and adapting the next weight $w_{j,i}(t_j + 1)$. Notice that

$w_{j,i}$ means the accumulated user's preference of keyword in the j th line. We calculate the similarity S_j^h between Q_{in} and the query field of j th line of the user's query history file, and similarity S_j^b between Q_{in} and the title field of j th line of the bookmark file.

$$S_j^h = \frac{w_{j,i} \cdot g(k_i)}{\sum_i w_{j,i} \cdot g(k_i)} \quad (2) \quad \& \quad S_j^b = \frac{w_{j,i} \cdot g'(k_i)}{\sum_i w_{j,i} \cdot g'(k_i)} \quad (3)$$

Where, $g(k_i)=1$ if $k_i \in Q_{in}$, otherwise $g(k_i)=0$, and $g'(k_i)=1$ if $k_i \in T_j$, otherwise $g'(k_i)=0$. Also, we calculate the similarity S_j^{url} between Q_{in} and the URL of j th line using equation (4).

$$S_j^{url} = \frac{s_{url}}{c_{in} + d_j \cdot s_{url}} \quad (4).$$

Where, $c_{in}=|Q_{in}|$, $s_{url}=|Q_{in} \cap URL_j|$, $d_j=|URL_j|$, and URL_j stands for the set of words in the URL of j th line. Weighting factor $0 < \alpha < 1$ is also defined by heuristics. Then, the total similarity between user's query and his/her query history file is calculated by the following equation, using a heuristic weighting factor α .

$$\arg \max_j (S_j^{url} + (1 - \alpha) \cdot S_j^h) \quad (5)$$

Second: By the same way, we calculate the total similarity between the user's query and his/her bookmark file, using a heuristic-weighting factor α :

$$\arg \max_j (S_j^{url} + (1 - \alpha) \cdot S_j^b) \quad (6)$$

2.5 IP Representation and Relevancy Measurement by WPA

IP is a vector of important terms, which is extracted and weighted by analyzing the contents of the Web page. Since the terms are not all equally important for content representation of IP vector of each WPA, an importance factor (.) is assigned to each term and decided by the kind of HTML tag, in which the term is included in the Web page. This means that WPA will emphasize/de-emphasize some keywords based on the value of .. The WPA calculates the weight of the term and constructs its IP vector from the number of appearance (tf) and the kind of tag, which includes the term within the Web page (e.g., in title, in header, in link, is bold, underline, italic,...) using equation (7).

$$w_{ik} = \frac{tf_{ik}}{\sum_k tf_{ik}} \quad (7) \quad \& \quad w_i = \frac{n}{k=1} w_{ik} \quad (8)$$

Where w_{ik} stands for the weight of term i in position k , and tf_{ik} stands for the number of occurrences that term i appears in position k specified by HTML tags. k stands for the weight decided by the kind of HTML tag $_k$ that includes the term i in the Web page. The total weight of a term i in the IP is the sum of all the weights in the HTML document of the Web page and is calculated by using equation (8).

Where n is the number of HTML tags within the Web page. The WPA $_i$ calculates the confidence factor that reflects the relevancy between the input query vector Q_{in} and its IP vector using the cosine coefficient equation.

$$Sim(Q_{in}, IP) = \frac{\sum_{i=1}^n h(k_i) w_i}{\sqrt{\sum_{i=1}^n h(k_i)^2} \sqrt{\sum_{i=1}^n w_i^2}} \quad (9)$$

Where, $h(k_i) = 1$ if $k_i \in Q_{in} \cap IP$, otherwise $h(k_i) = 0$.

2.6 Link Similarity with the Query and IP by WPA

The WPAs calculate the relevancy with the user's query based on the terms they have in both of their IPs and the hyperlink structures of the WPAs. This similarity function is based on both query-IP and query-link similarities. It is a hybrid similarity function that includes two components. The first component S_{ij}^{Q-link} measures the similarity between the user's query i and the URL of a Web page j and is calculated using equation (4). The second component S_{ij}^{Q-IP} measures the similarity between the user's query i and the IP of a WPA of Web page j and is calculated using the cosine coefficient equation (9). The whole similarity is calculated as follows. Where α is a heuristic-weighting factor $0 < \alpha < 1$.

$$S_{ij}^{Total} = (\alpha S_{ij}^{Q-Link} + (1 - \alpha) S_{ij}^{Q-IP}) \quad (10)$$

3 Implicit Response Implication by UIA

By observing user's behavior, it is possible to infer implicit feedback without requiring explicit judgments. Previous studies have shown that reading time to be a useful source of predicting UP implicitly [11]. We are investigating other sensors in correlation with the time of visiting the page to let Kodama's UIA detects the actual user's implicit response. These sensors are, the size of the selected page, the number of links within the page, as the user may try to visit some links within the page. Monitoring user's operations such as saving, printing, bookmarking, scrolling, minimizing, maximizing or closing the page. Jumping to another link, where the UIA distinguishes between two types of links in the Web page, if it is between pages with different or the same domain names. The selected page's server response, as the user may spend time and finally the server says it is unknown domain. Other heuristic factors like, type of the page (text, image, or applet), number of visits to this page, did the user bookmark, save or print this page? We defined hypotheses to infer the implicit response of the user based on these sensors.

4 Autonomous Adaptation in the WPA and UIA

Because we are automatically creating the IP of each WPA based on the contents of its corresponding Web page and creating the UP based on the user's interests, it is necessary to improve them autonomously. There are several approaches that can be used to learn a UP [1, 2, 3, 10]. The WPA allows only relatively small change of the weight of IP's terms based on the user's response, because adding/removing some terms into/from the IP may change the context of the Web page. When the user says the page is interesting, the WPA changes the weight of the terms in its IP, if and only if these terms appear in Q_{in} , in order to make better match with the query to be entered next time. If the user's query is $Q_{in} = \{q_1, q_2, \dots, q_n\}$, then the WPA checks if $q_i \in IP$ then changes its weight w_i by adding a reward value r_i to be $w_i + r_i$, else ignores it.

The UIA picks a number of keywords from the title and the headers of the selected document K_s in addition to the keywords of Q_{in} and creates a new list of keywords for the feedback K_f , where, $K_f = Q_{in} \cap K_s$. According to user's response, the UIA will modify the weight of the keyword in the UP using equation 1 and modify the number of visiting. If one of the keywords is not exist in the query field, then UIA adds it with an initial weight reflecting the user's response. The UIA refines the contents of the UP files by deleting the keywords that have weights less than a predefined threshold value. If the selected URL is not exist in the UP, then UIA adds a new record and initializes its query field. By this way, the UP will evolve over time to reflect the user's interests. Also, the keywords of the query and title fields continually moved closer to or away from their URLs.

5 Experimental Results

We have performed several experiments to make a consistent evaluation of Kodama system performance [7, 8]. The results we have obtained for users, who used the system from 10th of October until 10th of January, verify the facts that Kodama can learn and adapt to the user's preferences over time. Also, the idea of Web page agentification promises to achieve more relevant information to the user.

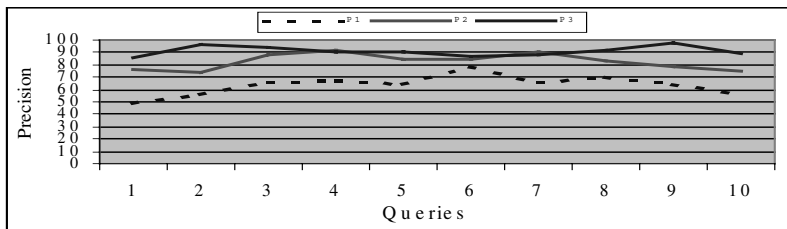


Fig. 2. Precision of the retrieved URLs to the user's queries to the IEEE domain

In the first experiments, we attempted to verify that the mechanism of agentifying the Web is useful for retrieving relevant information. We agentified several Web servers by giving the portal address of the Web servers to the system, the system creates the hyper structure of the WPA communities based on the hyperlink structure of each Web server. Figure 2 shows the Precision of the retrieved URLs to user's queries to the IEEE agentified domain <http://computer.org/>. The number of agentified Web pages in this Web server is about 2000. In the experiment,

1. We gave 10 ambiguous queries to the system, disabled the UIA's query expansion mechanism and calculated the Precision (P1) of the retrieved URLs to the given queries.
2. We allowed the UIA to expand the given query from the UP, then submitted the new query to the community of WPAs and calculated the Precision (P2) of the retrieved URLs to the expanded queries.
3. The UIA created the context query for the filtration then filtered the retrieved documents by comparing these documents with the context query and then we calculated the Precision (P3) of the filtered URLs to the context queries.

The results depicted in Figure 2 shows that the idea of Web page agentification, query expansion and filtration have been done by the UIA promise to achieve relevant information to the users and promoted using Kodama as a pinpoint IR system.

In the second experiments, we measured how well Kodama is being able to adapt to the user's interests over time and to get a good correlation between each URL and its relevant keywords. In order to understand the experiment, we define a Fitness value, which will show the correlation between the weights of keywords calculated by UIA and user's actual interest to each keyword, as follows.

- (1) User's actual interest: $S_j = \sum_{k=1}^m b_k W_k$, where W_k is the weight of keyword_k, $b_k = 1$ if the user judges keyword_k in the URL_j as relevant for his/her query, else $b_k = 0$.
- (2) Interest calculated by UIA: $T_j = \sum_{k=1}^m W_k$.

Then, we define the Fitness value $F_j = S_j / T_j$, which reflects the correlation between the two interests for URL_j. In the experiment, a user gave fifteen different queries, each of which consists of 1 to 5 keywords, then, after frequent interactions of retrieval, the user checked the relevancy of each keyword in the retrieved URL, then Fitness value was calculated for each URL in the UP. The Fitness values calculated after five and ten times retrieval interactions are shown in Figure 3. Figure 3 shows that the values of S and T are converging over time to each other, and this means that UIA is being able to predict and adapt to its user's interests.

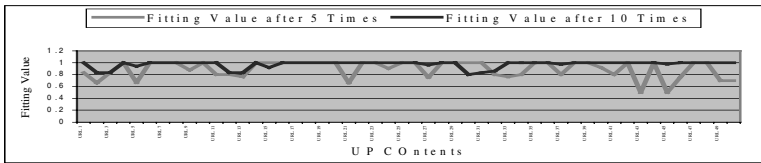


Fig. 3. Converging to the User's Interest over time

In the third experiments, we measured how well the feedback mechanism of UIA enables its user to access to more relevant information with very high relevancy. We define the Feasibleness of the feedback mechanism by M/N . Where N means the number of queries given by a user, and M means the number of the retrieved results at highest rank with which the user get satisfied. In the experiment, a user initially starts by giving a set of ambiguous and non-sense queries. At this point, URLs are retrieved and held in UP. Then, the system repeats the interaction process, in which the user gives a query, gives back the evaluation to the retrieved URLs, and the rank of URLs in UP is changed according to the response. In the experiment, the interactions were repeated ten times. Results of the experiments are shown in Figure 4. Figure 4 shows that the Feasibleness gradually goes up with time, and this means that UIA is helpful for users to retrieve relevant URLs.

In the fourth experiments, we measured how well Kodama system could adapt the contents of the UP in a way that reflects the user's preferences of each URL over time. We started with an UP contains different URLs about music. The user in this experiment gave eleventh queries about music. I.e., "I want to see information about Beethoven piano sonata concert." The UIA exploits the UP contents, shows the results to its user, receives user's response upon the relevancy of the retrieved URL with the given query and adapts the contents of UP. According to Figure 5, the contents of UP adapted to user's interests by increasing or decreasing the keyword's weights of some URLs in a way that reflects user's interest of these keywords with

the correlated URLs. Our approach was to change smoothly the user's preferences over time. This change was necessary to test if the system would be able to adapt in abrupt change as equilibrium the perfect alignment to the user preferences.

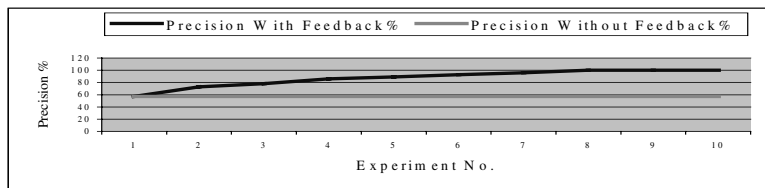


Fig. 4. Feasibleness while catching the user's behavior

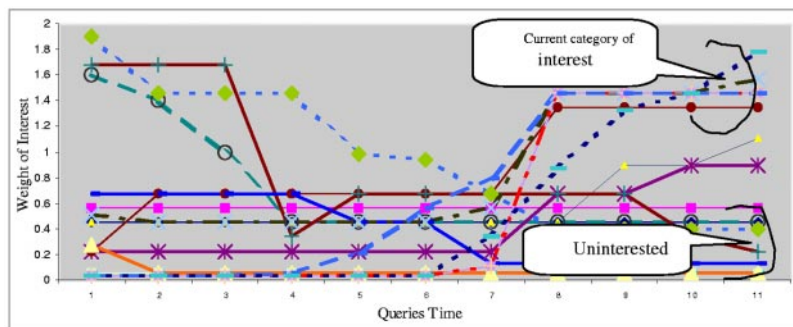


Fig. 5. Dynamic adaptation of the UP contents

6 Conclusion and Future Work

It is a straightforward idea to incorporate the idea of user modeling with machine learning methods into Web search services. We introduce Kodama system prototype, which is being developed and in use at Kyushu University, as a multi-agent-based approach to build an intelligent IR system that lets users retrieve more relevant distributed information from the Web. This paper explores ways to incorporate user's profiles into the search process to improve the search results. We reported methods to agentify the Web, and to exploit UP & IP adaptively on the Kodama system. We carried out several experiments to investigate the performance of our system. Through these experiments, we ensure that the idea of Web page agentification promises to achieve relevant information to the user. So, Kodama can be used as a pinpoint IR system that learns and adapts to the user's preferences over time. Future step in Kodama is extending our experiments in multiple SA domains and developing a smart query routing mechanism in the UIA for routing the user's query. Routing refers to the process of selecting the SA to be queried and forwarding queries to it. UIA will route the query to an appropriate SA, instead of sending the query to all SAs and gathering a large amount of noisy Web pages inconsistent with user's information need. In such a situation, the UIA and SA need an intelligent query routing mechanism that suggests the most relevant SA based on user's query history and some attributes of the SAs.

References

1. Ballacker K., S. Lawrence, and L. Giles, "CiteSeer: An Autonomous System for processing and organizing scientific Literature on the Web", Working notes of Learning from Text and the Web, Conference of Automated Learning and Discovery (CONALD-98), Carnegie, Mellon University, 1998.
2. Budzik J. and Hammond K. "Watson: Anticipating and Contextualizing Information Needs", in Proceedings of Sixty-second annual Meeting of the American Society for Information Science, 1999.
3. Chen Liren and Katia Sycara, WebMate: A Personal Agent for Browsing and Searching", Proceedings of the Second International Conference of Autonomous Agents, Minneapolis/ST, MN USA, May 9-13, 1998, pp.132-138.
4. Edmund S. Yu, Ping C. Koo, and Elizabeth D. Liddy: Evolving Intelligent Text-based Agents, Proceedings of the 4th International Conference of Autonomous Agents, June 3-7-2000, Barcelona, Spain, pp.388-395.
5. Helmy T., B. Hodjat and M. Amamiya, "Multi-Agent Based Approach for Information Retrieval in the WWW", Proceedings of the First Asia-Pacific International Conference on Intelligent Agent Technology (IAT99), Hong Kong, 15-17/12, 1999, pp. 306-316.
6. Helmy T., T. Mine, G. Zhong, M. Amamiya, "A Novel Multi-Agent KODAMA Coordination for On-line Searching and Browsing the Web", Proceedings of The Fifth International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents, 10-12/4, 2000, Manchester, UK, pp. 335-338.
7. Helmy T., T. Mine, G. Zhong and M. Amamiya, "Open Distributed Autonomous Multi-Agent Coordination on the Web", Proceedings of The Seventh International Conference on Parallel and Distributed Systems Workshops, pp. 461-466: July 4-7, 2000, Japan.
8. Helmy T., T. Mine and M. Amamiya, "Adaptive exploiting User Profile and Interpretation Policy for Searching and Browsing the Web on KODAMA System", Proceedings of the 2nd International Workshop on Natural Language and Information Systems NLIS, London, Greenwich, United Kingdom, September 4-8, 2000, pp. 120-124.
9. Hodjat B. and M. Amamiya, "Applying the Adaptive Agent Oriented Software Architecture to the Parsing of Context Sensitive Grammars", IEICE TRANS. INF. & SYST., VOL. E83-D, No.5 May 2000.
10. Joachims Thorsten, Dayne Freitag, and Tom M. Mitchell, WebWatcher: A tour guide for the World Wide Web, in Proceedings of International Joint Conference on Artificial Intelligence (IJCAI97), pp. 770-775, 1997.
11. Kim J., Oard D., and Romanik K. "Using Implicit Feedback for User Modeling in Internet and Intranet Searching" Technical Report [2000], Collage of Library and Information service, University of Maryland at Collage Park.
12. Kleinberg J., "Authoritative sources in a hyperlinked environment", ACM Journal, 46(s), PP. 604-632 1999.
13. Menczer F., A.E. Monge: "Scalable Web Search by Adaptive Online Agents: An InfoSpiders Case Study". Intelligent Information Agents, Springer, 1999.
14. Mladenec Dunja, "Text-learning and Related Intelligent Agents", IEEE Expert special issue on Application of Intelligent Information Retrieval, July-August 1999. pp. 44-45.
15. Pann K., A. And Sycara, K. "A Personal Text Filtering Agent", Proceedings of the AAAI Stanford Spring Symposium on Machine Learning and Information Access, Stanford, CA, March 25-27, 1996.
16. William G., S. Lawrence and C. Giles, "Efficient Identification of Web Communities", ACM Proceedings of KDD 2000, Boston, MA, USA.

Analyzing Workflow Audit Trails in Web-Based Environments with Fuzzy Logic

Gerald Quirchmayr¹, Beate List², and A. Min Tjoa²

¹ University of South Australia, School of Computer and Information Science
Mawson Lakes Boulevard, Mawson Lakes, SA 5095, Australia
Gerald.Quirchmayr@unisa.edu.au

² Vienna University of Technology, Institute of Software Technology
Favoritenstrasse 9 – 11 / 188, A-1040 Vienna, Austria
{list, tjoa}@ifs.tuwien.ac.at

Abstract. In web based environments it is essential to make the best possible use of what little information is available about a customer. That is why so many companies are trying to collect as much data from their customers during a visit on their web sites as possible. This data becomes even more valuable when combined with the audit trail of the logistics chain behind the web site. This paper describes an approach for applying Fuzzy techniques to the analysis of such audit trails with the goal of creating rules for better predicting customer behavior.

1. Introduction

In the past few years enterprises focus on creating process-centered organizations in order to reduce cost and improve customer satisfaction [3], [7]. Before the awareness of business process orientation companies were built up around products or services, resulting in gigantic functional hierarchies. Today, organizations cannot survive solely with marketing high quality products or services. Their success depends more and more on creative business models combined with efficient business processes (e.g. amazon.com). The execution of business processes has a huge impact on customer behavior: those processes that fail might cause complain procedures or even loose customers for ever and those processes that succeed might generate frequent customers and new, ‘productive’ processes. Therefore, a business process must not be seen as an independent entity: the integration of customers, generated revenue as well as initiated complain procedures must be considered for analysis. This comprehensive approach enables the classification of certain customer groups and their behavior in order to meet their requirements.

In this work we utilize the process warehouse concept, which has been developed by the authors and is a data warehouse especially designed for analyzing workflow audit data in order to create decision rules when information is incomplete or partially contradictory.

2. The Process Warehouse Approach

Workflow-based process controlling has received relatively little coverage in the related literature [20]. Research prototypes and commercial products in this area are built on top of relational databases and focus primarily on monitoring a small fraction of performance measures within a limited time frame. The analysis of workflow history, which is stored on instance state level, requires a lot of complex queries and transformations that cause a negative impact on the database performance. Operational workflow repositories store workflow histories only for a few months, but analysing data patterns and trends over time requires large volumes of historical data over a wide range of time. Several years of history would be useful for such analysis purposes. In order to avoid these shortcomings (discussed in [8], [9].) we apply a data warehouse approach, which is dedicated to analytical processing and which is well suited for fast performance in mining applications towards bottleneck diagnosis and other important decision supporting analysis.

The main objective of a performance measurement system (PMS) is according to Kueng in [10] to provide comprehensive and timely information on the performance of a business. This information can be used to communicate goals and current performance of a business process directly to the process team, to improve resource allocation and process output in terms of quantity and quality, to give early warning signals, to make a diagnosis of the weaknesses of a business, to decide whether corrective actions are needed, and to assess the impact of actions taken. A PMS ought to represent all goals and structures of an organisation. These are turned down into well-defined performance indicators, which are fundamental for the process warehouse design, as the analysis and improvement capability of the system depends highly on the integration of these aspects as well as on the transformation into an adequate data model.

Data from additional source systems e.g. Business Process Management Systems (target values, process definition), Enterprise Resource Planning Systems (organizational model, position profile, employee qualification, education, pay scheme), strategic data sources (balanced scorecard, key performance drivers and indicators) and other data sources (staff opinion surveys, customer surveys, special product offers, product announcements, stakeholder analysis, marketing and advertising events) lead to a very balanced and comprehensive performance measurement system. Our intent is to provide a decision support approach to business process control data and to exploit the analysis capabilities through the combination with business data. In this paper, we focus beside customer and order information on the workflow audit trail as the main data source.

We define the process warehouse (PWH) as a separate read-only analytical database that is used as the foundation of a process oriented decision support system with the aim to analyse and improve business processes continuously. It enables process analysts to receive comprehensive information on business processes very quickly, at various aggregation levels, from different and multidimensional points of view, over a long period of time, using a huge historic data basis prepared for analysis purposes to effectively support the management of business processes [12].

The analysis of business processes requires the representation of theoretical aspects in the basic concept, which we capture in four views [12]. The Business Process View completely disregards the functional structure, but fully represents the approach of process-centered organisations and looks horizontally across the whole organisation. The analysis of this view focuses on the process as a complete entity from a process owner's point of view. The process owner or manager is an individual concerned with the successful realisation of a complete end-to-end process, the linking of tasks into one body of work and making sure that the complete process works together [3], [7].

Business processes flow through several organisational units and cross a lot of responsibilities; it is obvious that the process reflects the hierarchical structures of the organisation [11]. The analysis of this view addresses the organisational structure of a business process and the fact that business processes, which cross organisational boundaries very often tend to be inefficient because of changing responsibilities or long delay times [11]. Therefore, the analysis of the organisational structure is an important aspect of process improvement, as it supports the detection of delay causing organisational units. The Organisational View supports the analysis of these aspects.

The Improvement Support View is based on the histories of several instances together. The aggregation of instances aims to identify major performance gaps and deviations, which give evidence of improvement needs. As single instances do not have an impact on aggregated performance, gaps reflect fundamental performance problems or process design shortcomings. The Information Detail View is targeting process, activity and work item information on instance level or slightly aggregated level. It enables the analysis of instance development over time and supports to determine the cause of performance gaps and deviations.

3. Audit Trails as Basis for the Analysis of Customer Satisfaction

Workflow Management Systems (WFMSs) carry out business processes by interpreting the process definition, which is designed in the build-time component of the WFMS. When customers, workflow participants or workflow engines create a new process instance, the workflow engine assigns an initial state to the workflow instance. Further, according to the process definition workflow engines create sub-process, activity or work item instances and assign work items to the process participant's work list. After completion of the work item it is withdrawn from the work list. Workflow participants can impact the pace of the workflow instance as they can explicitly request certain process instance states. Process definitions include very often process models with splits that depend on application data; therefore the workflow engine can access application data, called workflow related data.

Audit data, is the historical record of the progress of a process instance from start to completion or termination [16]. WFMSs store histories of all instances in log files, recording all state changes that occur during a workflow enactment in response to external events or control decisions taken by the workflow engine. In [17] a detailed specification of audit data can be found, but nevertheless an extension with process related data or other workflow control data is feasible.

The business process we define in the build time component of the WFMSs is a flight selling process of an airline on the web and is modeled in **Fig. 1** with the role activity diagram notation (see **13**). The business process starts when a customer enters the flight selling web page, chooses destination and date and checks the availability of the desired flight. As a result, a new workflow instance is initiated. A list of available flights is returned to the customer, who is expected to choose a flight-number and request fares. After the customer enters username and password, he or she can make reservations and purchase tickets online. Further, he /she is requested for travel purpose (business or leisure). This will create an order number and a booking code, which confirms the ticket. At the end of the flight selling process the customer is asked whether he or she was satisfied with the response time of the web service.

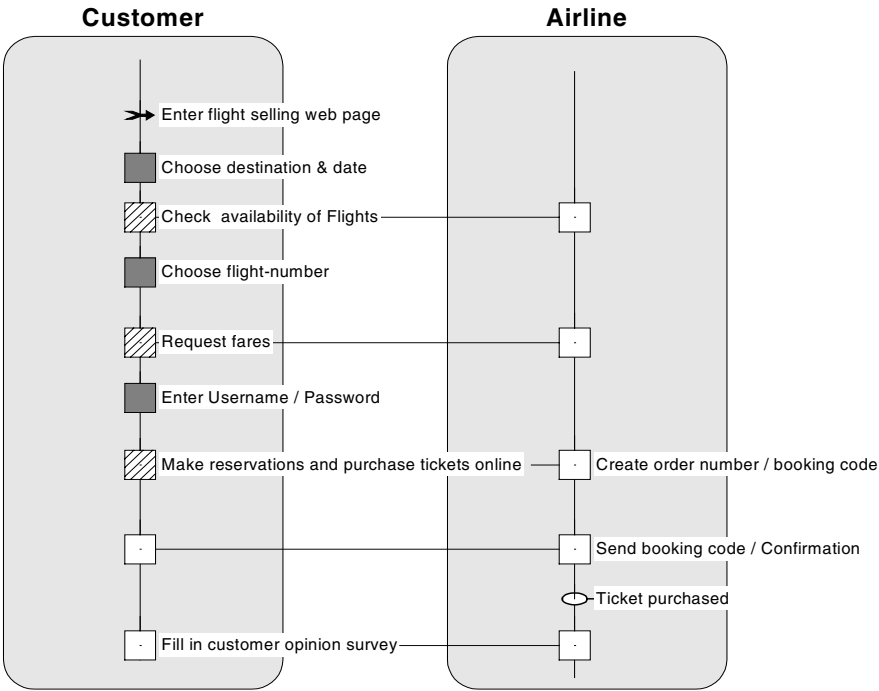


Fig. 1. Flight selling business process of an airline on the web

WFMSs track any workflow instance state change with a timestamp, therefore we can easily find out the duration of the workflow instance and calculate a deviation compared to the planned workflow duration in the process definition. In our example (see **Table 1**) we integrate the order number into the audit trail, as this extension enables the link with all available customer data, revenue and complain statements related to a certain order, booking code or workflow instance. The workflow engine retrieves the order number as process related data during the runtime of a workflow instance. As the order is via the web, we are also able to integrate the geographic part of the world where the process was initiated. In this example we combine the process

instance and the order number, as current commercial WFMSs do not support a customization of their audit trail. Research prototypes like Mobile [14](#) already address these shortcomings.

Table 1. Audit trail of the selling business process

CREATED	EVENT	ACTIVITY	PROCESS	INSTANCE	USER NAME	ACTIVITY NAME
		STATE	NAME	ORDER NO		
04.10.00 19:23	21007	21201	Flight_selling	264389456	CUSTOMER	Reservation
04.10.00 19:23	21010		Flight_selling	264389456	CUSTOMER	Reservation
04.10.00 19:23	21010		Flight_selling	264389456	CUSTOMER	Reservation
04.10.00 19:23	21031	21200	Flight_selling	264389456	CUSTOMER	Reservation
04.10.00 19:24	21007	21201	Flight_selling	264389456	ADMIN	Reservation
04.10.00 19:24	21012	21204	Flight_selling	264389456	ADMIN	Reservation
04.10.00 19:24	21034		Flight_selling	264389456	ADMIN	Reservation
04.10.00 19:25	21006	21200	Flight_selling	264389456	ADMIN	Confirmation
04.10.00 19:25	21007	21201	Flight_selling	264389456	ADMIN	Confirmation
04.10.00 19:25	21010		Flight_selling	264389456	ADMIN	Confirmation
04.10.00 19:25	21010		Flight_selling	264389456	ADMIN	Confirmation
04.10.00 19:25	21012	21204	Flight_selling	264389456	ADMIN	Confirmation
04.10.00 19:25	21025		Flight_selling	264389456		Confirmation

4. Expanding the Capacities of the Process Warehouse with a Formal Model for Forecasting with Incomplete Data

The envisaged analysis process draws on lessons learned in data warehouses and artificial intelligence. Creating the rules driving knowledge-based systems has always been a major challenge and solving this problem is a key requirement for making the systems maintainable. Our process does therefore consist of three main stages:

1. The collection of data during the workflow execution of processes
2. The extraction of workflow audit trails into the process warehouse and their enrichment
3. The creation of rules based on enriched audit tails

The major advantage of this approach is that workflow audit trails can be transformed via the process warehouse directly into rules that allow us to forecast the satisfaction of customers. Once these rules are created, they can be used to indicate how a certain type of customer will react to a given environment. A typical example are airline web sites, where it could for example be useful to know which type of customer cares most about a quick response of the system in order to be able to give queries launched by this type of customer the necessary priority.

Assuming that audit trails are created in certain intervals and that a new rule base is automatically created from this latest information, this process can also be viewed as learning system. It can adapt itself to the actually prevailing requirements of customers instead of basing its reaction on a set of static rules.

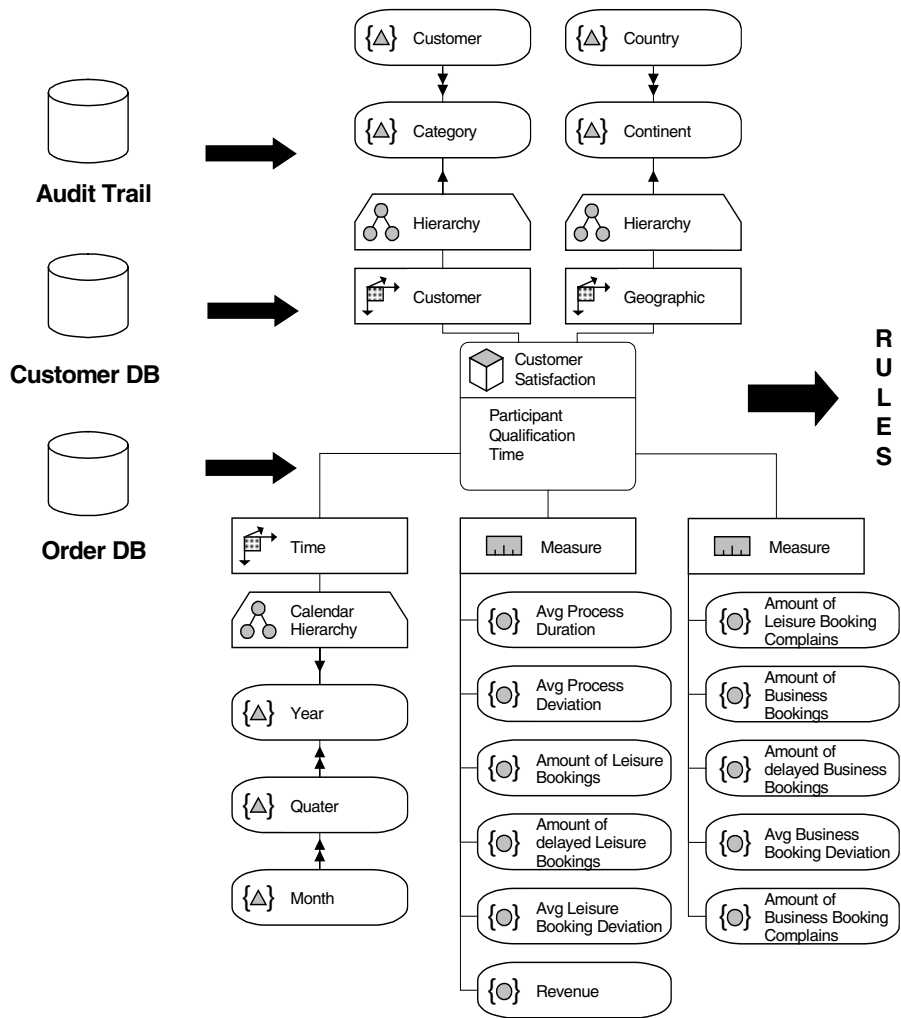


Fig. 2. The Analysis Process Model (Customer Satisfaction Cube in ADAPT notation)

5. Application to the Forecasting of Airline Customer Satisfaction

The enriched audit trail created in section 4 can now be used as input for the analysis phase of our model. Typically, a leisure and even more so a business traveler books

with more than one airline so that no single airline will ever have a complete booking history. Consequently approaches have to be followed which allow for reasoning from incomplete data. That is why the analysis part of the process was based on the concept developed in [4]. For background information the reader is referred to the following related work: [1], [2], [4], [5], [6], [13], [18] and [19].

Following this formal model we can show how fuzzy certain and possible rules can be constructed from the audit trail. In most situations it might be difficult to determine the exact membership value. Assuming that an airline customer is a 30% leisure and a 70% business traveler would be the traditional approach. A customer service department may however only be able to produce a slightly vague information, i.e. that the customer is with a belief of between .2 and .4 a leisure and with a belief between .6 and .8 a business traveler. This uncertainty can also lead to a certain amount of imprecision in the conclusion(s) derived from these attributes. Analyzed on a quarterly basis an audit trail produced from a logistics workflow in an online booking system might therefore lead to the conclusion that during the first quarter a customer was 20% leisure and an 80% business traveler. In the second quarter this might change to 40% leisure and 60% business travel. In a traditional statistical model the simple average would be taken, while a Fuzzy approach allows us to model the customer as leisure traveler with a belief between .2 and .4 and as business traveler with a belief between .6 and .8. Other attributes, such as the duration of the booking process on the web site can also be modeled in the same way. In general this leads to a fuzzy membership in a subinterval of [0, 1].

As described in [4], such a set can symbolically be written as $A = \langle a_i, b_i \rangle / x_i$, indicating that A is a fuzzy set whose value at x_i is the interval $\langle a_i, b_i \rangle$, also called Fuzzy Set of Type II. A very natural way of generalizing this case are the following definitions:

$$I - A = \langle I - b_i, I - a_i \rangle / x_i$$

$$\overline{Max}\{\langle a_i, b_i \rangle, \langle a_j, b_j \rangle\} = \langle Max(a_i, a_j), Max(b_i, b_j) \rangle$$

$$\overline{Min}\{\langle a_i, b_i \rangle, \langle a_j, b_j \rangle\} = \langle Min(a_i, a_j), Min(b_i, b_j) \rangle$$

From the above formulas we can construct the definition of operators I and J . Specifically, if A and B are fuzzy sets whose values are subintervals of [0,1], then

$$\overline{I}(A \# B) = \overline{MinMax}_x(I - A(x), B(x)) \text{ and } \overline{J}(A \# B) = \overline{MaxMin}_x(A(x), B(x)).$$

According to [4] it should be noted that $1 - A$ is an interval as is $B(x)$. Thus $\overline{Max}(I - A(x), B(x))$ is some interval depending on x and $\overline{I}(A \# B)$ then is some interval (independent from x). The same observation applies to $\overline{J}(A \# B)$. Thus the degree to which A is a subset of B , or more accurately, the degree to which A implies B , is interval-valued as is the degree to which A possibly implies B .

We now demonstrate the technique in an example where we have two attributes: a web sites response time to a customer's request (quick or slow) and the type of trav-

eler (leisure or business). The customer service analyst can then conclude whether a customer will be satisfied or not. The goal in this example is to find out whether it is essential for a certain type of customer that the web site responds quickly.

Table 2. Forecasting technique applied to flight selling business process on the web

0	Response of Web Site	1Type of Traveler	2Customer Satisfaction
x_1	$\langle .2, .4 \rangle$ /slow + $\langle .6, .8 \rangle$ /quick	$\langle .1, .3 \rangle$ /leisure + $\langle .7, .9 \rangle$ /business	$\langle .6, .8 \rangle$ /satisfied + $\langle .2, .4 \rangle$ /upset
x_2	$\langle .3, .5 \rangle$ /slow + $\langle .5, .7 \rangle$ /quick	$\langle .8, .9 \rangle$ /leisure + $\langle .1, .2 \rangle$ /business	$\langle .7, .9 \rangle$ /satisfied + $\langle .1, .3 \rangle$ /upset
x_3	$\langle .6, .9 \rangle$ /slow + $\langle .1, .4 \rangle$ /quick	$\langle .5, .7 \rangle$ /leisure + $\langle .3, .5 \rangle$ /business	$\langle .5, .6 \rangle$ /satisfied + $\langle .4, .5 \rangle$ /upset
x_4	$\langle .2, .3 \rangle$ /slow + $\langle .7, .8 \rangle$ /quick	$\langle .1, .2 \rangle$ /leisure + $\langle .8, .9 \rangle$ /business	$\langle .8, .9 \rangle$ /satisfied + $\langle .1, .2 \rangle$ /upset
x_5	$\langle .7, .8 \rangle$ /slow + $\langle .2, .3 \rangle$ /quick	$\langle .1, .2 \rangle$ /leisure + $\langle .8, .9 \rangle$ /business	$\langle .3, .4 \rangle$ /satisfied + $\langle .6, .7 \rangle$ /upset
x_6	$\langle .1, .4 \rangle$ /slow + $\langle .6, .9 \rangle$ /quick	$\langle .1, .2 \rangle$ /leisure + $\langle .8, .9 \rangle$ /business	$\langle .7, .9 \rangle$ /satisfied + $\langle .1, .3 \rangle$ /upset

From Table 2 construct the following pairs of Fuzzy Sets:

$$\text{Slow} = \langle .2, .4 \rangle / x_1 + \langle .3, .5 \rangle / x_2 + \langle .6, .9 \rangle / x_3 + \langle .2, .3 \rangle / x_4 + \langle .7, .8 \rangle / x_5 + \langle .1, .4 \rangle / x_6$$

$$\text{Quick} = \langle .6, .8 \rangle / x_1 + \langle .5, .7 \rangle / x_2 + \langle .1, .4 \rangle / x_3 + \langle .7, .8 \rangle / x_4 + \langle .2, .3 \rangle / x_5 + \langle .6, .9 \rangle / x_6$$

$$\text{Leisure} = \langle .1, .3 \rangle / x_1 + \langle .8, .9 \rangle / x_2 + \langle .5, .7 \rangle / x_3 + \langle .1, .2 \rangle / x_4 + \langle .1, .2 \rangle / x_5 + \langle .1, .2 \rangle / x_6$$

$$\text{Business} = \langle .7, .9 \rangle / x_1 + \langle .1, .2 \rangle / x_2 + \langle .3, .5 \rangle / x_3 + \langle .8, .9 \rangle / x_4 + \langle .8, .9 \rangle / x_5 + \langle .8, .9 \rangle / x_6$$

$$\text{Satisfied} = \langle .6, .8 \rangle / x_1 + \langle .7, .9 \rangle / x_2 + \langle .5, .6 \rangle / x_3 + \langle .8, .9 \rangle / x_4 + \langle .3, .4 \rangle / x_5 + \langle .7, .9 \rangle / x_6$$

$$\text{Upset} = \langle .2, .4 \rangle / x_1 + \langle .1, .3 \rangle / x_2 + \langle .4, .5 \rangle / x_3 + \langle .1, .2 \rangle / x_4 + \langle .6, .7 \rangle / x_5 + \langle .1, .3 \rangle / x_6$$

The interpretation of these equations is straightforward. For example, it says that x_1 is a poorly representative example of Slow rated between .2 and .4, while being far more representative of Quick since the rating is between .6 and .8.

We now compute intersections of attributes as

$$\text{Slow} \cap \text{Leisure} = \langle .1, .3 \rangle / x_1 + \langle .3, .5 \rangle / x_2 + \langle .5, .7 \rangle / x_3 + \langle .1, .2 \rangle / x_4 + \langle .1, .2 \rangle / x_5 + \langle .1, .2 \rangle / x_6$$

$$\text{Slow} \cap \text{Business} = \langle .2, .4 \rangle / x_1 + \langle .1, .2 \rangle / x_2 + \langle .3, .5 \rangle / x_3 + \langle .2, .3 \rangle / x_4 + \langle .7, .8 \rangle / x_5 + \langle .1, .4 \rangle / x_6$$

Similarly, expressions for Quick \cap Leisure and for Quick \cap Business can easily be computed.

From $I - A = \bigvee_i \langle I_{b_i}, I_{a_i} \rangle / x_i$ and $\overline{\text{Max}}\{\langle a_i, b_i \rangle, \langle a_j, b_j \rangle\} = \langle \text{Max}(a_i, a_j), \text{Max}(b_i, b_j) \rangle$ we obtain

$$\overline{\text{Max}}(I - \text{Slow} \quad \text{Leisure}, \text{Satisfied}) = \langle .7, .9 \rangle / x_1 + \langle .7, .9 \rangle / x_2 + \langle .5, .6 \rangle / x_3 + \langle .8, .9 \rangle / x_4 + \langle .8, .9 \rangle / x_5 + \langle .8, .9 \rangle / x_6.$$

From $\overline{I}(A \quad B) = \overline{\text{MinMax}}(I_{A(x)}, B(x))$ it follows that

$$\overline{I}(\text{Slow} \quad \text{Leisure} \quad \text{Satisfied}) = \overline{\text{MinMax}}(I_{\text{Slow} \quad \text{Leisure}}, \text{Satisfied}) = \langle .5, .6 \rangle.$$

Similarly,

$$\overline{\text{Max}}(I - \text{Slow} \quad \text{Business}, \text{Satisfied}) = \langle .6, .8 \rangle / x_1 + \langle .8, .9 \rangle / x_2 + \langle .5, .7 \rangle / x_3 + \langle .8, .9 \rangle / x_4 + \langle .2, .3 \rangle / x_5 + \langle .6, .9 \rangle / x_6.$$

Thus $\overline{I}(\text{Slow} \quad \text{Business} \quad \text{Satisfied}) = \langle .2, .3 \rangle.$

The corresponding extracted rules then read as follows:

*if the response time of the web site is slow **and** the customer is a leisure traveler **then** the customer will be satisfied [.5 .6];*
*if the response time of the web site is slow **and** the customer is a business traveler **then** the customer will be satisfied [.2 .3];*

In the crisp case, an equivalence class is the union of intersections of the form (Response time of web site = v_1) and (Type of traveler = v_2) where v_1 and v_2 are possible values of these attributes. The above format therefore truly is a generalization of the concept of equivalence classes generated by combinations of attribute classes. One should resist making any inference, for example, as to how much a slow response time implies a satisfied (respectively upset) customer (see [4]).

We now use $\overline{J}(A \# B) = \overline{\text{MaxMin}}(A(x), B(x))$ to compute possible rules.

For example, $\overline{\text{Min}}(\text{Slow} \quad \text{Business}, \text{Satisfied}) = \langle .2, .4 \rangle / x_1 + \langle .1, .2 \rangle / x_2 + \langle .3, .5 \rangle / x_3 + \langle .2, .3 \rangle / x_4 + \langle .3, .4 \rangle / x_5 + \langle .1, .4 \rangle / x_6$ and

$$\overline{J}(\text{Slow} \quad \text{Business} \# \text{Satisfied}) = \overline{\text{MaxMin}}(\text{Slow} \quad \text{Business}, \text{Satisfied}) = \langle .3, .5 \rangle.$$

Similarly $\overline{\text{Min}}(\text{Slow} \quad \text{Leisure}, \text{Satisfied}) =$

$$\langle .1, .3 \rangle / x_1 + \langle .3, .5 \rangle / x_2 + \langle .5, .6 \rangle / x_3 + \langle .1, .2 \rangle / x_4 + \langle .1, .2 \rangle / x_5 + \langle .1, .2 \rangle / x_6 \text{ and}$$

$$J(\text{Slow} \quad \text{Leisure} \# \text{Satisfied}) = \overline{\text{MaxMin}}_x(\text{Slow} \quad \text{Leisure}, \text{Satisfied}) = \langle .5, .6 \rangle.$$

The corresponding possible rule can then be written as:

*if the response time of the web site is slow **and** the customer is a business traveler
then the customer will possibly be satisfied [.3 .5];*
*if the response time of the web site is slow **and** the customer is a leisure traveler
then the customer will possibly be satisfied [.5 .6];*

In the present example one would of course have additional rules by considering Quick and Leisure, Quick and Business, and also by considering upset customers as well as satisfied ones. To reduce the number of rules, we may consider a threshold $0 < \alpha < 1$ and not use rules where the right ends of the corresponding intervals fall below α . When introducing the extracted rules into a knowledge-based system, the rules can serve as input for the knowledge acquisition system, the intervals can be used to determine the firing strength of the respective rules.

6. Conclusion

This paper has shown how rule extraction can be implemented by extending business process warehouses, which store audit trails with mechanisms for rule generation. The main contribution the paper tries to make is to show how ideas from workflow management, data warehouses and formal logic can be integrated for improving the capabilities of decision support systems. Once these rules are extracted they can serve as basis for letting a decision support system operate in two modes, a truly risk averse one which only makes use of *certain* rules and a more adventurous one which also makes use of *possible* rules.

References

1. Arciszewski, T., Ziarko, W.: Adaptive expert system for preliminary engineering design. Proc. 6th International Workshop on Expert Systems and their Applications, Avignon, France, 1(1986) 696-712
2. Cheeseman, P.: Induction of models under uncertainty. Proc. ACM SIGART Internat. Symposium on Methodologies for Intelligent Systems, Knoxville, Tennessee, (1986)
3. Davenport, T. H.: Process Innovation – Reengineering Work through Information Technology, Harvard Business School Press, Boston (1993) 1-17
4. Korvin, A. de, Quirchmayr, G., Hashemi, S., Kleyale, R.: Rule Extraction Using Rough Sets when Membership Values are Intervals. Proc. of the Ninth International Workshop on Database and Expert Systems Applications (DEXA 98), IEEE Press (1998)
5. Fibak, J., Slowinski, K., Slowinski, R.: The application of rough set theory to the verification of indications for treatment of duodenal ulcer by HSV. Proc. 6th International Workshop on Expert Systems and their Applications, Avignon, France, 1(1986) 587-599

6. Gryzmala-Busse, J.W.: Knowledge acquisition under uncertainty – rough set approach, *Journal of Intelligent and Robotic Systems* 1(1988) 3-16
7. Hammer, M.: *Beyond Reengineering*, Harper Collins Publishers (1996) 3-17
8. Inmon, W. H.: *Building the Data Warehouse*. Wiley & Sons (1996)
9. Kimball, R.: *The Data Warehouse Toolkit: Practical Techniques For Building Dimensional Data Warehouse*. John Wiley & Sons (1996)
10. Kueng, P., Wettstein, Th., List, B.: A Holistic Process Performance Analysis through a Process Data Warehouse. To appear in *Proceedings of the American Conference on Information Systems* (2001)
11. Leymann, F., Roller, D.: *Production Workflow – Concepts and Techniques*. Prentice Hall PTR (2000)
12. List, B., Schiefer, J., Tjoa A M., Quirchmayr, G.: Multidimensional Business Process Analysis with the Process Warehouse. In: W. Abramowicz and J. Zurada (eds.): *Knowledge Discovery for Business Information Systems*, Chapter 9, Kluwer Academic Publishers (2000) 211 – 227
13. Ould, M.: *Business Processes - Modelling and Analysis for Re-engineering and Improvement*. John Wiley & Sons (1995)
14. Schlundt, M., Jablonski, S., Hahn, C.: *Application Driven History Management for Workflow Management Systems*, Technical Report, Department of Computer Sciences (Database Systems), Universitaet Erlangen-Nuernberg, (2000)
15. Strat, T.M.: Decision analysis using belief functions. *Int. J. of Approximate Reasoning* 4(1990) 291-417
16. Workflow Management Coalition, Interface 1 – Process Definition Interchange, 1999, located at <http://www.aiim.org/wfmc/standards/docs.htm>
17. Workflow Management Coalition, Interface 5 – Audit Data Specification, 1998, located at <http://www.aiim.org/wfmc/standards/docs.htm>
18. Yager, R.R.: Approximate reasoning as a basis for rule based expert systems, *IEEE Trans. On Systems, Man and Cybernetics* 14(1984) 635-643
19. Zadeh, L.A.: The rule of fuzzy logic in the management of uncertainty in expert systems, *Fuzzy Sets and Systems* 11(1983) 119-227
20. Muehlen, M. zur, Rosemann, M.: Workflow-based Process Monitoring and Controlling – Technical and Organizational Issues, *Proceedings of 33rd Hawaii International Conference on System Sciences*, IEEE Press (2000)

Using Hypertext Composites in Structured Query and Search

Zhanzi Qiu, Matthias Hemmje, and Erich J. Neuhold

GMD-IPSI, Darmstadt, Germany

{qiu, hemmje, neuhold}@darmstadt.gmd.de

Abstract. This paper proposes a structured query and search model for applying link-based hypertext composites that can be represented with new Web standards in Web searching and describes a primitive prototype system that implements the model. We argue that by enabling users to query different levels of the composite structures with same or different keywords and getting search hits that are not separate nodes but sets of inter-linked nodes, the precision of the search results can be improved. Meanwhile, users may get more contextual information about the search results.

1 Introduction

For several years the World Wide Web Consortium (W3C) [23] has been working hard to create and promote base technologies for enabling the „semantic Web“ [2]. Among their efforts, XML (Extensible Markup Language) [21] and RDF (Resource Description Framework) [17] are most important. They improve the ability of expressing structures and semantics on the Web. However, it is still an open question how to use the structural and semantic information that is representable with these new Web standards efficiently for searching the Internet and filtering and retrieving relevant information.

This paper describes a part of our effort to answer the question. In this work, we focus on making use of hypertext composites, especially link-based hypertext composites, to formulate structured queries and derive structured search results. We propose that by enabling users to query different levels of the structures with same or different keywords and getting search hits that are not separate nodes but sets of inter-linked nodes, the precision of the search results can be improved. Furthermore, users can get more contextual information about the search results.

The remainder of the paper is organized as follows. Section 2 describes the concept of hypertext composites. Section 3 answers the question how hypertext composites can be represented with new Web standards. Section 4 proposes a schema for using hypertext composites in Web searching. Section 5 presents a prototype system designed for implementing the schema. Section 6 mentions related work. Finally, Section 7 summarizes this work and outlines our future activity.

2 Hypertext Composites

In a hypertext system, a *node* provides the „wrapping“ for a document or piece of information. A *link* represents a relation between nodes. The logical structures in hypertexts are usually supported by means of composites (groups of nodes and links) [11].

There are two kinds of hypertext composites: composite nodes that are composed by non-linking mechanisms (such as hierarchical structures in the structured documents defined by a DTD); and link-based composites that are composed by computation based on link types that represent containment (or part-of) relations. In most cases, especially in a single system, non-linking composition mechanisms are more efficient than link-based composition mechanisms [4]. However, there are also many cases where link-based constructs are desired [10].

This work is to explore the value of link-based hypertext composites in Web searching. Thus unless clearly specified, composites later in this paper refer to link-based ones. A more formal description about such composites is first given below.

2.1 Formal Description of Link-Based Hypertext Composites

A *link-based hypertext composite* is a special kind of node that is constructed out of other nodes and composites [11]. These nodes and composites are *components* of the composite. They are linked from the composite or the other components in the composite with containment (or part-of) relation. They may also link to each other with other types of relation. No link-based hypertext composite may contain itself either directly or indirectly.

Precisely, if C is a *link-based hypertext composite*, then its contents must contain a pair (N, L) , where N is a set of nodes in a hypertext graph and L is a set of semantic links whose endpoints belong to N . For any $n_1 \in N$, there exists $link(C \rightarrow n_1) \in L$ and $link(C \rightarrow n_1).type$ represents containment relations, or, there exists $n_2 \in N$ so that

$link(n_1 \rightarrow n_2) \in L$ AND $link(n_1 \rightarrow n_2).type$ represents containment relations,
OR

$link(n_2 \rightarrow n_1) \in L$ AND $link(n_2 \rightarrow n_1).type$ represents containment relations.

We say that C contains a node M if M is in N and that C contains a link l if l is in L . M is a *node component* of C , while l is a *link component* of C .

Usually the *components* of a *hypertext composite* refer to its *node components*, as the meaning of the link components is mostly reflected in the building-up process of the composite. This usage of *components* is adopted later in this paper.

2.2 Hypertext Composites and Hypertext Contexts

Based on the above definition of hypertext composites, a hypertext composite can be seen as a special kind of hypertext context. A *hypertext context* is a generic high-level hypermedia structure that groups together a set of nodes and links into a logical whole [15]. It can be used to combine nodes from one or more hypertext composites to describe a specific view of the nodes in a hypertext system.

2.3 A Simple Example

One main use of a link-based hypertext composite is to organize a document in a hierarchical structure. A typical example is an online user manual, which has hierarchical structure as its backbone, but also has other hypertext links within or across the document boundary.

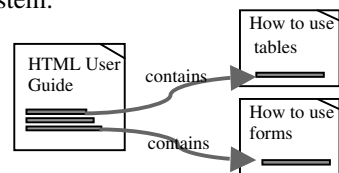


Fig. 1. A simple composite „HTML User Guide“

Figure 1 shows a simple hypertext composite „HTML User Guide“ which contains the nodes „How to use tables“, „How to use forms“ and so on as its components. These nodes can be grouped into various hypertext contexts that represent the different views of the document that is organized hierarchically with the composite.

3 Representation of Hypertext Composites

The traditional Web has only a single node type called the page. All pages are equally accessible in a „flat“ pool [10]. Something like the effect of composites can be obtained using pages full of URLs, but true structuring composites are not supported.

This situation has been changed due to some new Web standards. HTML 4.01 [12] has defined a kind of LINK element which may only appear in the HEAD section of a document to describe relations between documents and a set of link types permitted in the documents. It also allows users define additional link types, by using a meta data profile to cite the conventions used to define the link types. This makes it possible for an application system to compute link-based composites based on the link types.

The LINK element defined in HTML 4.01 contains a *rel* attribute to specify the relationship of the linked document with the current document and a *rev* to describe a reverse link from the linked document to the current document. The value of the both attributes is a space-separated list of link types. The predefined link types that relate to the containment relation between Web documents and can be used to compute hypertext composites are *contents*, *chapter*, *section* and *subsection*.

For instance, in the Figure 1, suppose the document for „How to use tables“ is „table.html“ and the document for „How to use forms“ is „form.html“. They both are chapters of a document collection for „HTML User Guide“, whose table of contents is in „HTML.html“. Then „HTML.html“ may contain the following exemplary encoding for describing the containment relation:

```
<HEAD> ... other head information...
<LINK rel="chapter" href="table.html">
<LINK rel="chapter" href="form.html">
</HEAD>
```

Or, „table.html“ and „form.html“ may contain:

```
<HEAD> ... other head information...
<LINK rel="contents" href="HTML.html">
</HEAD>
```

With respect to XML documents, XLink [20] provides an efficient mechanism to represent typed links. In XLink, links are encoded in linking elements. The types of links can be encoded via the *role* attribute of linking elements. The values of this attribute are of a kind of CDATA. They may be predefined in DTDs (fixed) or specified in documents (no default value is provided in DTDs). For instance, suppose in the Figure 1 the XML document for „HTML User Guide“ is „HTML.xml“, the document for „How to use tables“ is „table.xml“ and the document for „How to use forms“ is „form.xml“. The following out-of-link extended xlink can be contained in „HTML.xml“ to describe the simple composite shown in the figure:

```
<content xml:link="extended" inline="false">
  <locator href="table.xml" role="contains">
    <locator href="form.xml" role="contains">
  </content>
```

Still, RDF [17] provides a more systematic way to describe the composites, whose components may be either HTML documents or XML documents or any other kinds

of Web resources. For instance, the following sample RDF encoding uses Dublin Core [7] vocabularies to describe the simple composite shown in Figure 1 (suppose the documents are in HTML format):

```
<rdf:RDF xmlns:rdf="http://www.w3.org/TR/WD-rdf-syntax#"
  xmlns:dc="http://purl.org/metadata/dublin_core#"
  xmlns:dcq="http://purl.org/metadata/dublin_core_qualifiers#">
  <rdf:Description about="HTML.html">
    <dc:Relation>
      <dcq:RelationType
rdf:resource="http://purl.org/metadata/dublin_core_qualifiers#hasPart"/>
      <rdf:value resource="table.html"/>
      <rdf:value resource="form.html"/>
    </dc:Relation>
  </rdf:Description>
</rdf:RDF>
```

The link type „hasPart“ is represented by the tag <dc:Relation> and the rdf:resource attribute in <dcq:RelationType>. In fact, Dublin Core has defined two qualifiers for containment relations: „has Part“ and „is Part of“. They provide an ideal way to represent hypertext composites.

4 Using Hypertext Composites in Structured Query and Search

The possibility of representing hypertext composite information with the new Web standards leads us to explore new search methods making use of the information. We suppose that based on composite structures, hyperstructure-based query and search facilities can be implemented. These facilities will enable users to query different levels of the structures with the same or different keywords (see the left-hand side part of Figure 2) and get search hits that are not separate nodes but sets of inter-linked nodes (see the right-hand side parts of Figure 2). Compared to the search results that are single nodes, the structured search results may be more precise and relevant to users' some specific information needs. Furthermore, users will be provided with more contextual information about the nodes contained in the results.

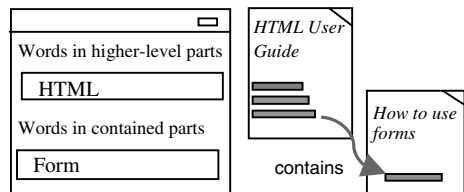


Fig. 2. Structured query and search using link-based composite structure

4.1 Structured Queries Based on Hypertext Composites

Formally, the general structured queries based on hypertext composites can be described as follows:

Definition 1 *Structured Query Levels*

Structured query levels STQLs is a value that indicates how many structural levels are to be contained in a structured query.

Definition 2 *Query Terms*

A query term is a keyword or phrase used to search for nodes indexed in a document collection.

$QT = \{QT_i\}$, where $1 \leq i \leq N_{qt}$, N_{qt} is the number of query terms

Example: $QT = \{< \text{HTML user guides}>, <\text{digital library}>, <\text{information retrieval}>\}$

Definition 3 *Qualifiers*

A qualifier is used to describe the quality and form of the query terms that users input.

$Q = \{Q_i\}$, where $1 \leq i \leq N_q$, N_q is the number of qualifiers

Example: $Q = \{<=>, <^*?>\}$, here „ $=$ “ means exactly like, „ * “ means using stemming expressions

Definition 4 *Logical Operators*

A logical operator is used to combine logically two query terms or avoid (negation) a query term.

$L = \{L_i\}$, where $1 \leq i \leq N_l$, N_l is the number of logical operators

Example: $L = \{ , , \neg \}$ where \wedge means AND; \vee means OR; \neg means NOT.

Definition 5 *Level Query Expressions*

A level query expression \overline{LQ}_l is a content descriptor used to search for nodes in the structural level l . It is constructed by one or more query terms QT combined by logical operators L.

$$\overline{LQ}_l = (QT_1 L_1 QT_2 \dots L_{k-1} QT_k),$$

where $k \geq 1$, QT_i QT, Qualifier (QT_i) Q, L_j L, $1 \leq j \leq k-1$

Definition 6 *Structured Query Expressions*

A structured query expression is a conjunction of some level query expressions.

$$\overline{SQ} = (\overline{LQ}_1, \overline{LQ}_2, \overline{LQ}_3, \dots, \overline{LQ}_n),$$

where $n = \text{STQLs}$, i.e. the structured query levels, „ \wedge “ is logical operator AND

Example: $\overline{SQ} =$ („User guide“, „HTML“, „form“), which searches for a three-layer hyperdocuments with „User guide“ in the first level, „HTML“ in the second level, „form“ in the third level.

4.2 Structured Search Results Based on Hypertext Composites

As mentioned, the structured search hits resulted from the structured queries are not separate nodes but sets of inter-linked nodes. That is, in each search hit, links representing containment relations exist between the nodes.

Formally, the search results for a structured query \overline{SQ} can be described with the following definitions:

Definition 7 *level Query Results*

The level query results for a level query expression (\overline{LQ}_i) are a set of nodes that „contain“ or „not contain“ the query terms (with the qualifier as quality control) specified in \overline{LQ}_i .

$$LQR(\overline{LQ}_i) = LQR(QT_1 L_1 QT_2 \dots L_{k-1} QT_k) = \{N_i\}$$

where N_i „contains“ or „not contains“ (with Q as qualifier) QT_j QT

Definition 8 *inter-containment-linked node chains*

An inter-containment-linked node chain is a series of nodes, each of which (except the first one) is linked from the node before it with a type that represent containment relations and (except the last one) also links to the node after it with a type that represent containment relations.

$$CILNC = \{CILNC_i\}$$

$$CILNC_i = \{N_1(\text{contains}) \rightarrow N_2(\text{contains}) \rightarrow \dots \rightarrow N_n\},$$

where n = the number of the nodes in the chain

Definition 9 *Structured Search Results*

The structured search results for a structured query expression (\overline{SQ}) are a set of inter-containment-linked node chains, which are derived by computing the

containment links between the nodes contained in the level query results corresponding to the level query expressions in \overline{SQ} .

$$SSR(\overline{SQ}) = SSR(\overline{LQ_1}, \overline{LQ_2}, \overline{LQ_3}, \dots, \overline{LQ_n}) = \{CILNC_n\}$$

$$CILNC_i = \{N_1(\text{contains}) \rightarrow N_2(\text{contains}) \rightarrow \dots \rightarrow N_n\},$$

where N_i LQR($\overline{LQ_i}$), $n = \text{STQLs}$, i.e. the required structured query levels

4.3 Issues for Supporting Structured Query and Search Based on Hypertext Composites

Representing hypertext composite information in standard ways (as described in Section 3 above in this paper) is the prerequisite to enable the use of the information in structured queries and searches. In addition to this prerequisite, a search system that intends to implement the search method should at least be able to

- gather from the Web available hypertext composite information by computing the links between the nodes (pages) and organize the information in the system efficiently,
- provide a friendly, adaptive interface to enable users to specify structured queries in a comfortable way,
- derive the structured search results with acceptable system performance, and
- present structured search results in a way that is good for users to understand and get more contextual information about the results.

In the next section we will see how our prototype system addresses these issues.

5 A Prototype System

The prototype system is designed for testing our idea of using hypertext composites in structured query and search and studying the feasibility of the model proposed above. It contains the components that are mostly layered and represent different aspects (as shown in Figure 3).

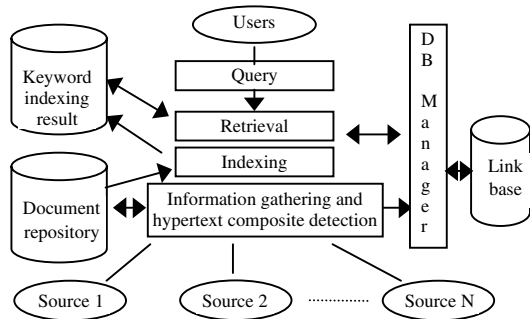


Fig. 3. High-level system architecture

5.1 Information Gathering and Hypertext Composite Detection

The information gathering and hypertext composite detection component in the system performs mostly 2 functions: Web crawling and hypertext composite detection. Web crawling is to download the Web resources (HTML, XML and RDF docs, whose URIs are given by a URI server) specified by users or parsed out from the resources during the processing and stores them in the document repository. Hypertext composite detection is to read the repository and parse the documents in it. Every web resource gets an associated ID number called a nodeID that is assigned whenever a new URI is parsed out of a web resource. All link information is extracted and stored (through a store server) in the link base of the system. The database schema for the link base is shown in Figure 4.

Node Table	<table><tr><td>nodeID</td><td>URI</td></tr></table>	nodeID	URI		
nodeID	URI				
Link Type Table	<table><tr><td>linktypeID</td><td>linktype</td></tr></table>	linktypeID	linktype		
linktypeID	linktype				
Link Table	<table><tr><td>linkID</td><td>source_nodeID</td><td>linktypeID</td><td>target_nodeID</td></tr></table>	linkID	source_nodeID	linktypeID	target_nodeID
linkID	source_nodeID	linktypeID	target_nodeID		

Fig. 4. Database for hypertext composites – a link base

With the link information in the link base, any composites in the collection can be computed dynamically when necessary and be used to derive structured search results that meet users’ structured queries.

5.2 Indexing and Retrieval

The indexing component in the system is to do keyword indexing to the Web resources (HTML and XML docs) gathered. The keyword indexing technology is so mature that we do not need to give much detail about it. As an example, our prototype system simply makes use of Glimpse [8] as its keyword indexing and search engine.

Then the structured search results are derived in this way: the retrieval component of the system first sends query terms in each structured query level to Glimpse and gets distinct responding results, and then uses the link information stored in the link base as filters to get structured search results.

For instance, the process to derive the structured search results for the exemplary structured query with 3 levels

$$\overline{SQ} = (\text{“User guide“}, \text{“HTML“}, \text{“form“})$$

is as shown in Figure 5.

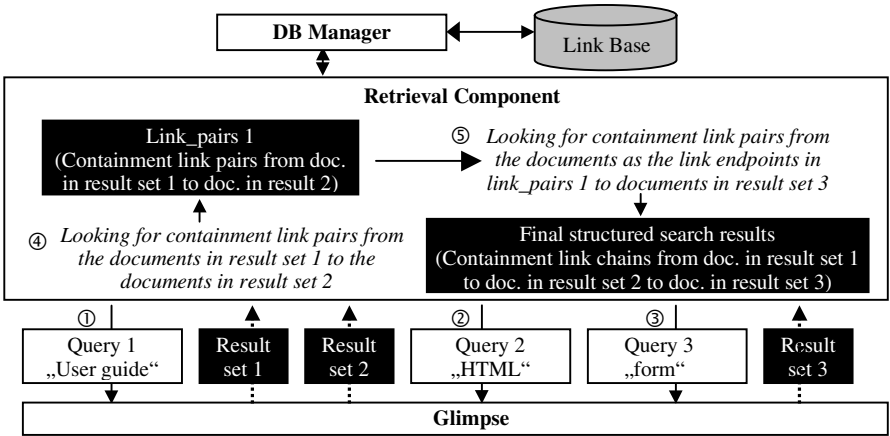


Fig. 5. Deriving structured search results (an example)

In the figure, the *result set 1* is a set of documents that contain „User guide“. The *result set 2* is a set of documents that contain „HTML“. The *result set 3* is a set of documents that contain „form“. The *link_pairs 1* is set of link pairs. In each link pair the „from“ node belongs to the result set 1, the „to“ node belongs to the result set 2. Finally, the *final search results* are a set of link chains. In each chain, the first node

belongs to the result set 1, i.e. contains the term „User guide“. The second belongs to the result set 2, i.e. contains the term „HTML“. The third belongs to the result set 3, i.e. contains the term „form“. The link pairs or link chains are constructed based on the links of the types that represent containment relations in the link base.

5.3 Adaptive Form-Based Interface for Formulating Structured Queries

The query component in the system is responsible for enabling users to specify structured queries, transferring the queries to the retrieval component, and presenting search results derived by the retrieval component to users. It is implemented as Web browser clients with an adaptive form-based user interface. The adaptivity of the interface is mainly reflected in the adjustability of structured query levels. That is, if the user first selects 2 levels but gets unsatisfied results, he/she may ask the system to adjust the structured query levels to 3 or more. Every time when the value of the structured query levels is modified, the system will regenerate the form. An exemplary query in such an interface is shown in Figure 6.

5.4 Presenting Structured Search Results

As described, the structured search results derived from the structured queries based on hypertext composites are not separate nodes but sets of inter-linked nodes. How to present them to users is also crucial for the system's success. A good presentation may improve users' satisfaction to the results and enable users get more contextual information about the results and even the relevant resources.

In the moment the prototype system just presents the structured search results in a simple but integrated way. A table, which contains the designated structured query levels as the number of columns, is built to contain the results. Each row represents one result. A screenshot for an exemplary result presentation is given in Figure 7.

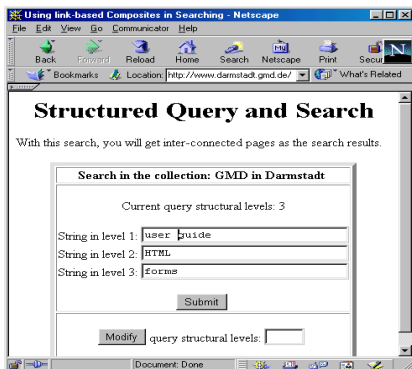


Fig. 6. Form-based interface for formulating structured queries

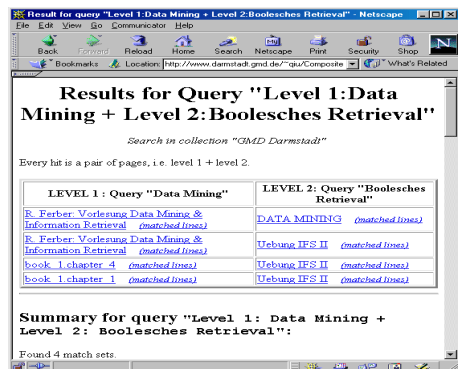


Fig. 7. Presenting structured search results

5.5 Evaluation Issues

A thorough evaluation about the system should measure not only the quality of its search results but its storage requirements and various performances as well. We have not done such a thorough evaluation but performed some primary experiments, in

which we aim to provide users more specific search results when they query about the technical documents, online user manuals, teaching materials and other kinds of well-organized hyperdocuments in the GMD Darmstadt site (<http://www.darmstadt.gmd.de/>). We are sure that the system will provide users more precise search results that meet users' some specific information needs and more contextual information about the results in the whole collection than normal search systems that provide users single pages as search results. Any application domains that own rich hypertext composite information represented in the standard ways will benefit from supporting the proposed search method.

There is also some significance that the system can scale well to the size of the Web, as it chooses a scalable DBMS (Informix Universal Server) and stores all link information gathered for computing the hypertext composites in databases.

6 Related Work

There is a trend of making use of additional structural information to improve Web searching. Structural information (mainly links) has so far been used for enhancing relevance judgements [1; 9; 14; 24; 18; etc.], ranking Web pages [5; 13; 3; etc.] or other purposes [e.g. 19]. However, it keeps an open question how to use the structural and semantic information that is representable with new Web standards (mainly XML [21] and RDF [17]) efficiently for searching the Internet and filtering and retrieving relevant information.

To answer this question, we have proposed a schema for searching in the Web space by using hypertext contexts as search boundaries [15] and an idea of making use of link-based domain models, which are hyperstructures that have domain specific semantics, in formulating structured queries [16]. As hypertext composites can be seen as a special kind of hypertext contexts, the schema for using hypertext contexts in searching is also applied to hypertext composites.

This work is a part of our most recent effort to answer the question. The structured query and search model we propose is for making use of link-based hypertext composites, while the XML Query [22] is to provide flexible query facilities to extract data from XML documents, in which the composite nodes are composed by non-linking mechanisms.

Finally, with respect to the structures used in search and the form of the search results, the structured search we mean in this work is different from the general structured search in the domain of database technology, which is the traditional domain that structured search tools fall into. A very good introduction to database technologies is [6].

7 Summary and Future Work

This work proposes a structured query and search model for applying link-based hypertext composites that can be represented with new Web standards in Web searching and describes a primitive prototype system that implements the model. We argue that by enabling users to query different levels of the composite structures with same or different keywords and getting search hits that are not separate nodes but sets

of inter-linked nodes, the precision of the search results can be improved. In addition, users may get more contextual information about the search results.

The follow-up work will be to perform a thorough evaluation about the method and the system and to study how the proposed schema can be used effectively in practice as more hypertext composite information is provided on the Web. Based on the evaluation result, further improvements to the method and the system will be done. A more intuitive visualized interface for formulating structured queries and presenting structured search results will be implemented in the system, and other alternative query execution plans might be considered.

It will also be done to explore the value of the composite information in page ranking, filtering and the other activities related to Web searching. For instance, the appearance of a page at a higher level in the structures would be ranked higher when providing single pages as search results in a system.

References

1. Arocena, G. O., Mendelzon, A. O., Mihaila, G. A., „Applications of a Web query language,” *Proc. 6th International World Wide Web Conference*, 1997.
2. Berners-Lee, T., „What the semantic web can represent,” available at: <http://www.w3.org/DesignIssues/RDFnot.html>.
3. Brin, S. and Page, L., „The anatomy of a large-scale hypertextual Web-search engine,” *Proc. 7th International World Wide Web Conference*, 1998.
4. Carr, L., Hill, G., Roure, D. D., Hall, W., and Davis, H. „Open information services,” *Proc. 5th International World Wide Web Conference*, 1996.
5. Carriere, J., Kazman, R., „WebQuery: Searching and visualizing the Web through connectivity,” *Proc. 6th International World Wide Web Conference*, 1997.
6. Chris, J. Date. *An Introduction to Database Systems*. The Systems Programming Series. Addison-Wesley, Reading, Massachusetts, sixth edition, 1995.
7. Dublin Core: <http://purl.oclc.org/dc/>
8. Glimpse: <http://glimpse.cs.arizona.edu/>
9. Golovchinsky, G., „What the query told the link: the integration of Hypertext and Information Retrieval,” *Proc. 8th ACM Conference on Hypertext*, pp. 67-74, 1997.
10. GRONBAK, K. and TRIGG, R., „Towards a Dexter-based model for open hypermedia: Unifying embedded references and link objects,” *Proc. ACM Hypertext'96*, pp.16-20, 1996.
11. Halasz, F. and Schwartz, M., „The Dexter Hypertext Reference Model,” *Communications of the ACM*, 37(2), pp.30-39, February 1994.
12. HTML 4.01 Specification (W3C Recommendation 24 December 1999). Available at: <http://www.w3.org/TR/html4/>.
13. Kleinberg, J., „Authoritative sources in a hyperlinked environment,” *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 1998.
14. Marchiori, M., „The quest for correct information on the Web: Hyper search engines,” *Proc. 6th International World Wide Web Conference*, 1997.
15. Qiu, Z., Hemmje, M., Neuhold, E. J., "ConSearch: Using hypertext contexts as Web search boundaries," in: Etzion, Opher & Scheurmann, Peter (Eds.), *International Conference on Cooperative Information Systems (CoopIS 2000)*, (pp. 42-53). Berlin [etc.]: Springer, 2000 (Lecture notes in computer science; 1901).
16. Qiu, Z., Hemmje, M., Neuhold, E. J., „Using Link-Based Domain Models in Web Searching,” *Proc. 2000 KYOTO International Conference on Digital Libraries: Research and Practice*, November 13-16, 2000, Kyoto University, Kyoto, Japan.
17. RDF: <http://www.w3.org/RDF/>

18. Rivlin, E., Botafogo, R. and Shneiderman, B., „Navigating in hyperspace: designing a structure-based toolbox,“ *Communications of the ACM*, 37(2), pp. 87-96, 1994.
19. Spertus, E., „ParaSite: Mining structural information on the Web,“ *Proc. 6th International World Wide Web Conference*, 1997.
20. XLink: <http://www.w3.org/TR/xlink/>.
21. XML: <http://www.w3.org/XML>
22. XML Query: <http://www.w3.org/XML/Query>
23. W3C: <http://www.w3.org/>
24. Weiss, R., Veles, B., Sheldon, M., Nemprenpre, C., Szilagyi, P., Gifford, D. K., „HyPursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering,“ *Proc. 7th ACM Conference of Hypertext*, 1996.

Categorizing Distribution Model Scenarios for Online Music

Willms Buhse

Dept. of General and Industrial Management,
Technical University of Munich, Germany,
and
Bertelsmann Digital World Services, New York
Willms.Buhse@dwsco.com

Abstract. This paper will examine the potential distribution models in the electronic market for online music. The goal of this paper is to categorize distribution models according to several case studies involving new distribution mechanisms like file sharing, streaming and super distribution. The virtualization of music leads to challenges on the supply and demand side. On the supply side, the main question is if the distribution of online music can be controlled by the offering party. On the demand side, the consumer preference for buying digital goods is being analyzed. Are consumers willing to pay directly for digital goods or will the revenue be collected indirectly by other public or private entities? As a result, distribution models for online music can be categorized into four scenarios. In the first scenario, online music is used to promote the traditional business of the CDs. Therefore penetration pricing and lack of additional security features are used to allow maximum distribution of the music. In the second scenario, again music can not be protected but consumers are willing to pay for a service that enhances their experience and makes it easier to browse, listen to and collect music. The third scenario is significantly different from the first two scenarios as music labels are expected to be able to protect their rights by technical means. Using digital rights management technology that allows to control the distribution of music files using encryption or watermarking, and revenues can result from subscription services. In the fourth scenario peer to peer technology allows consumers to use a mechanism called super distribution with which they can share and recommend songs. The paper concludes with a recommendation to music labels to position themselves in all four scenarios.

1. Introduction

This paper will examine the potential distribution models in the electronic market for online music.

Online music is defined as commercially available digital music that is distributed over networks like the internet. Thereby music has become the ideal case study for digital commerce with its unique availability in digital form on billions of CDs. Additionally, the existence of compression technologies like MP3, combined with

growing bandwidth, allows for mass market consumption of high-quality music. The music industry, though small in its market size, has become a prominent case study for new technology concepts, introduced by companies like Napster for peer-to-peer file sharing, RealNetworks for streaming media, InterTrust for digital rights management and others.

Forecasts from analysts regarding the market size for online-music, vary significantly between 7.8bUS \$ (Forrester), 2.6b US \$ (Jupiter) and 1.9b US \$ (Market Tracking International).[1]

Though much literature can be found prognosticating a significant change in the competitive environment of the music industry, little research exists on the combination of revenue models and property rights in online music.[2]

Digital Music

The music industry adopted digital media long before any other mass media industry with the development of the synthesizer in 1960. With the advent of the synthesizer and further development with the MIDI standard in 1982, the way music was composed, produced and performed was revolutionized.[3]

On the consumer side, the digital era began in 1970 with the invention of the Compact Disc (CD). It was first introduced by Sony and Philips in 1980 and soon became the most popular medium for storage and distribution of digital music.

With the introduction of the internet, end consumers started to use networks for the delivery of music instead of physical media. From the beginning online distribution became an underground phenomenon.[4] The catalyst for this phenomenon was the use of compression technology developed by the Fraunhofer Institute in Erlangen in the early 90s. This compression technology is referred to as MP3, a compression format that reduces the quantity of data by using psycho-acoustic mechanisms.

Although music always was considered an information good, on electronic markets, the impact is much greater than for digital music stored on physical media.

2. Electronic Markets for Online-Music

The business of digital goods is considered the “heart of electronic commerce”, with increasing importance for modern economics.[5] Early literature centers around various e-commerce theories within the framework of Transaction Cost Economics. The main hypothesis is the Electronic Market Hypothesis (Malone et al., 1987; Malone et al., 1989).[6] According to Malone, network technology will create a new, transparent market space in which buyers and sellers can be matched fast and at minimal costs. In electronic markets, producers would be able to cut overheads and eliminate traditional elements of the distribution chain, while consumers were to benefit from unlimited choice and decreasing costs.

Regarding changes in the value chain, new functions arise, which is defined in current literature as *intermediaries*. Typical functions of an intermediary are providing information about offer and demand, facilitate aggregation and distribution, and establish a trusted relationship. Additionally, secondary functions include payment transaction processing, financing etc.[7]

The starting point for this analysis is the assumption that the basic principle of the electronic market as efficient allocation mechanism works. But challenges on both the supply and demand side of the electronic market are leading to insufficiencies.

In the following, two significant consequences regarding the distribution models caused by the virtualization of music are analyzed: first the cost structure for the delivery is structured differently and thereby revenues might be affected. Second the protection of copyrights has become more difficult in today's networks.

According to Forsa, the majority of the internet users (69 percent) in Germany are not willing to pay for information or entertainment on the net.[8] One reason that may limit the willingness to pay for online-music may lie in the loss of a physical representation of the artist's work, which has become a collectible good with comprehensive artwork associated with it.[9] As a result, the internet seems to have a significant impact on the music industry's revenue model. Using file sharing, even distribution costs are shared among consumers which might make it difficult for companies to rationalize any price point.[10]

In the literature, revenues are divided into two main categories: *direct revenues* which result from the consumer, and *indirect revenues* which are refinanced through associated products via public or private entities.[11] While in the literature a separation between different revenue streams is possible, in the business environment, a spectrum of these streams can be found just like a news paper might have revenue streams from advertising, subscription and single transactions.

On the supply-side, the theory of public goods holds that goods have different characteristics whether or not there is rivalry or non-rivalry in using them. *Public goods* are *non-excludable* and *non-rivalrous* in consumption while private goods are sold to those who can afford to pay the market price. With non-excludable online-music, end consumers become *free riders*, who are not willing to pay the market price for music as long as others might be accessing the music for free.[12]

Traditionally the distribution of music is dominated by the major labels resulting from their oligopoly with five dominating players on the supply side. As a result, the music industry shows interest in privatizing the music in order to generate higher revenues not only from traditional products but also from the online market. From a technology point of view, the music industry started the Secure Digital Music Initiative (SDMI) to develop specifications jointly with technology companies like Microsoft, IBM and many others. These specifications are related to watermarking and encryption technologies. In the literature, many doubt that the music industry can successfully introduce security mechanisms that are either unbreakable or at least can raise the barrier for piracy without creating unproportional high costs.[13] Many examples in other media industries like currently the DVD-protection scheme have shown failures of secure protection mechanisms.[14] Additionally, on the internet only a single copy (even by re-digitizing from analog versions) made available is sufficient to be globally distributed in a short period of time leading to a total loss of control by the owner. But it is quite possible that the biggest challenge the music industry is facing is not hackers but instead infrastructure. Today's infrastructure with 200m multimedia PCs, 1b CD- audio-devices and 17b unprotected audio CDs with 150.000 different titles will be very difficult to replace.[15]

3. Distribution Model Scenarios

The goal of using scenarios is to categorize various distribution models like file sharing, digital rights management and super distribution.

As described in the previous chapter, the virtualization of music has two significant consequences regarding the distribution models: first the cost structure for the delivery is structured differently and thereby revenues might be affected. Second the protection of copyrights has become more difficult in today's networks.

By combining these two crucial issues into a matrix representing both supply and demand, four scenarios can be deduced.

Assumptions

These four distribution model scenarios are subject to the following assumptions:

- in the mid- to long-term, no distribution models will be viable which infringe on copyright laws. However, there might be systems without commercial interest that face no legal consequences for enabling illegal copies.
- revenue models are based on rational entrepreneurial decisions, excluding artistic, voluntary or otherwise motivated scenarios. Nevertheless, the variety of music, especially produced by independent musicians that so far did not have the opportunity to reach a global audience might benefit from lower marketing and distribution costs.
- Most importantly, these scenarios anticipate a slow migration towards online technologies. Meaning, traditional media companies maintain distribution control over physical storage media like CD and DVD.

3.1 First Scenario: File Sharing

Within less than two years, Napster became the largest music library ever with about 1b titles, without economic incentive, marketing activities, and even more important without involvement of the music industry.[16] At a very high level, file sharing systems or peer-to-peer-networks (P2P) aggregate and distribute information. With either central or de-central listings, files be can searched for, transferred and stored locally. The main challenge for content owners is its mass phenomena. Since its launch, Napster attracted almost 50 Million users who knowingly violate copyright laws.

While Napster through its partnership with Bertelsmann plans membership fees and the compensation of content owners, other open-source-file-sharing systems are developed without any commercial purpose. Their purpose is to freely distribute information beyond any control. Examples are Gnutella developed by Gene Kan and FreeNet designed by Ian Clarke. Both are designed to run de-centralized, which makes it almost impossible to control or shut down their operations. As a result, besides music files, other illegal content like children pornography and terrorist instructions can be found.

How can the music industry embrace such systems to generate revenues? Revenues can be generated indirectly from online music in return for the value of consumers' attention.[17] This can be used to promote either the physical album version or the artist in order to reach more popularity and thereby earn higher merchandising and advertising revenue. As a result, with online music being a public good, the

combination of online and offline business by integrating online-music and traditional marketing and distribution seems a profitable business model.[18]

Despite legal battles from RIAA arguing that illegal copies cannibalize album sales, market studies are inconclusive at this point. Jupiter identified Napster usage as one of the most important factor for increased music purchases.[19] On the other hand, VNU found album sales decreasing in record stores close to universities, where file sharing supposedly reaches high usage among students.[20]

Since the mid 90s, artist like Tom Petty or the “Toten Hosen” have promoted their albums with free downloads. Creed offered their hit song in 1999 from 100 web sites for free download, and in the process stimulated their album sales. Coincidentally their album “Human Clay” reached the top of the billboard charts.[21] A recent example is the partnership between the online retailer CDNow and Napster, where the file sharing system receives a commission of about 15% for every album sale.

Shapiro/Varian suggest the usage of online-versions to promote the hard copy, like the publishers “National Academy of Sciences Press” and “MIT Press” did by publishing online *versions* of their books on web pages with inherently less reading comfort, which resulted in the doubling of their offline sales.[22] The “Follow the Free”-pricing strategy can be used to achieve maximum reach with the prospect that consumers buy the advertised product.[23] The configuration of the version’s parameters thereby is the critical success factor. Regarding online music, this might be portability, ubiquity and especially compatibility with existing hardware. The traditional practices of collecting and gift giving music are not solved with digital goods and might result in long-term existence of traditional hardcopy media products. Nevertheless, substitution of traditional media like CDs and DVD-Audio might increase as soon as a comparable infrastructure for online music exists. Physical goods have always served as “containers” for services. For example, a CD has no intrinsic value, only the value of delivering music to your ears. In the age of downloadable music, though, the CD loses its value as a container for music.[24]

3.2 Second Scenario: Personalized Distribution Service

Provided online music is a public good, collecting direct payments seems almost impossible unless, the value lies primarily in the distribution, rather than in the content itself.[25]

In this scenario, instead of copy protection, service-oriented distribution models are developed that prevent the motive to copy. Besides content, these services offer convenience, reliability and fast access to music almost anywhere and at anytime and are referred to as the *celestial jukebox*. [26] This services sector is expected to grow from 2.5m today to 12.3m in 2003 in the U.S.[27]

Ultimately, those companies would have to combine content, community, application services, context and search functionality. Therefore, personalization plays a crucial role in attracting consumers and providing lock-in.[28] Traditionally, radio stations and music label could only satisfy a broad audience, while the internet allows a segment between mass media and individual communication between 5000 to 10000 listeners.[29] In the networked economy, these versions and even individual products and services are achievable due to smaller transaction and production/service costs.[30]

Using a feedback loop mechanism for online-music, personal playlists can be generated, recommended, updated and shared among other users. Online music, with about three min title length can generate comprehensive sets of data over time, provided 4 hours of daily music consumption, 80 songs might be rated on a daily basis almost automatically. Large description data bases like Moodlogic or Gigabeat can analyze relationships among titles and artists according to rhythm, instruments, contextual information and even mood.

It might be easy to maintain a piracy site with some illegal copies, but to provide access, payment mechanisms and customer service to many thousand people simultaneously is a more complex task.[31] Which companies might position themselves in the role of music service provider? First, relationships, such as those established by radio or television stations, emphasize repeat visits. They have already proven their ability for selection and aggregation of music.[32] Second, those with existing billing and services relationships like ISPs and TelCos, e.g. AOL TW. Third, technology companies with proprietary technology for music browsing, storage and delivery, artist aggregation, etc. For example, locker companies offer access to purchased music from anywhere, like Myplay or BeamIT. In the fourth group, there are companies with a link to end devices, like hardware-, OS-software-, and CE-device-manufacturers, though they might as well bet on copy protection technologies as they are able to choose and set standards.

Nevertheless, under current copyright law, most companies might have to negotiate licenses either directly with the music labels, their syndication partners or through royalty collecting entities, in order to be able to offer these services.

3.3 Third Scenario: Distribution Using Subscriptions and Memberships

Protection technologies play an important role in determining whether a media product is a public or a private good.[33] In scenarios three and four, online music is considered a private good, as content owners are able to restrict access to the content and thereby introduce the possibility to exclude free riders and charge for their online music.

To securely protect online music, all major labels have incorporated *digital rights management* technology, which basically falls into four categories: first the *access* is controlled at the users right of entry with passwords, encryption and/or authentication. Second, the *usage* is controlled according to rules that are set by the distributor of the music. This determines how the user can interface with the information, e.g., listen-only rights, where the user is unable to save or distribute the music. Third a *tracking* mechanism allows the information provider to track subsequent use with watermarking and digital footprints. Fourth and last, *payment* systems enable the information provider to generate revenue for the rights granted to the user. As a result of inefficient micro payment systems, subscription models are viewed as a method to overcome high transaction costs.[34]

For subscription models watermarking can provide important contributions to the field of intellectual property protection within a more extensive security framework for identification and proof of ownership, which is comparable to IRSC-Codes used by the GEMA for recognition of CD-Audios.[35] By embedding a watermark into the compressed audio signal during delivery, the customers are aware that a watermark may identify them.[36] Hence, they can be made responsible if the signal is found

outside the legal domain by a trigger technology, even in a decompressed and analog representation.[37] In contrast to encryption technologies, watermarks could be used with today's infrastructure for CD-Audio as well as MP3-devices.

With subscription models, large numbers of information goods can be distributed for a fixed price. In a variety of circumstances, a multi-product monopolist can extract substantially higher profits by offering one or more bundles of information goods than by offering the same goods separately.[38] At the same time, bundling can be used to introduce new artists and titles as a strategy to overcome the information paradox, which states that the value of an information can't be determined *a priori* of consumption.[39]

In this scenario, for the first time in their history, music labels have the opportunity to create a continuous relationship with the end consumer. This relationship offers a foundation on which music labels can generate revenues. The subscription model may represent a mix between indirect and direct revenues with the option of consumption combined with transparent pricing.[40] A premium membership might offer a flatrate, eventually combined with services from the second scenario, while an advertising-based membership might limit access in quantity, time or actuality.

3.4 Fourth Scenario: Super Distribution

In 1990, a visionary architecture was developed for the distribution of digital goods by the Japanese Ryoichi Mori, who coined the term *Super distribution* for a new concept of licensing information.[41] The fundamental idea is to allow free distribution and copy of information, while controlling access to changes and usage with the owner defining the terms. According to his prototype, called Software Service System (SSS), which was implemented as a peer-to-peer-architecture, the following components must be available:[42]

- a persistent *cryptographic wrapper* must stay in place when the digital property is used, copied, redistributed, etc.
- a *digital rights management system* with a trusted tool that tracks the deals and the usage associated with the access to the digital property
- the *payment information* have to be exchanges securely among the parties

After securely encrypting the music with a key, the package can be digitally delivered to the consumer's end devices.[43] There, the locally installed trusted tool gains access to the digital content with an unlock key which leaves the file locally encrypted and streams the digital content into the memory for "on the fly" decryption. The user who has agreed to the terms and conditions of use, has now the license to access the content. His usage is recorded and the transaction is reported to a clearinghouse to initiate payments and backup system information. Using the super distribution concept, consumers can recommend and share files among each other via email, FTP, physical media and even file sharing networks. Still the copyright is being protected and the content owner maintains control and determines payment collection.

Under the third scenario, bundling was mentioned as being attractive for content companies to extract higher profits. In the music industry, this has always been the case with album sales, where only one or two hits from an entire album initiate the purchase. Digital products possess optimal de-bundling capability which in return can be re-bundled again for custom-mixes.[44]

With digital downloads and super distribution, consumers might start “cherry picking” their hits and thereby endanger the traditional revenue model of album sales. At the same time, super distribution can be used to market content and benefit from the *lead-time-advantage* where information has higher value with minimal time lag.[45] The value of online music, might be highest close to the release date, where according to the windowing concept, highest profits can be made.[46]

In this scenario, using digital rights management and super distribution, major labels maintain control over the distribution of music and might even be able to enforce their copyrights more than in the traditional world.

4. Conclusion

In this paper I examined scenarios for online distribution models that depend on changes to the supply and demand side of the music industry. As a result, consistent distribution models in all four scenarios were developed. The scenarios have shown that there is a spectrum of potential revenue streams for online music both as public and private goods. Therefore, the main distinction between the scenarios depends on the supply side, where copyright for online music can either be protected by technical means or not.

Although online music distribution has been in place for some time, it is too early to determine which scenarios will evolve. Nevertheless, it is quite possible that the scenarios co-exist under certain market conditions. In this case, it is assumed that all four scenarios can come into affect during the life-cycle of an online music release. Starting with the secure super distribution concept (scenario 4) at the time of release, followed by a time-lag for subscription based distribution (scenario 3). Over time, the value might decrease and with hackers distributing illegal copies, the release might become widely accessible as a public good. Then services might be offered (scenario 2) and at the same time additional value from the user’s attention for promotion and advertising might be extracted (scenario 1).

Therefore, the music labels should prepare themselves to claim strategic positions in all four scenarios, otherwise their traditionally dominant role in the music market, and the barrier-to-entry that currently prevents external competition will diminish.

References

- [1] Becker, A., Ziegler, M.: Wanted: A survival plan for the music industry – Napster and the consequences. Diebold Study, Munich (2000)
- [2] Zerdick, A. et al.: Die Internet-Ökonomie – Strategien für die digitale Wirtschaft. Springer, Berlin, Heidelberg (1999)
- [3] Evans, P., Wurster T.: Blown to Bits - How the new economics of information transforms strategy. Harvard Business School Press, Boston, Massachusetts (1999)
- [4] Pettauer, Richard: Die Blitzkarriere von MP3. Konferenzbeitrag: Micafocus 1: Reales Muskschaffen für einen virtuellen Markt am 18. März 2000, viewed at http://www.mica.at/mf_pettau_p.html, 10-10-2000
- [5] Picot, A., Reichwald, R., Wigand, R.: Die grenzenlose Unternehmung, 7. Auflage, Wiesbaden (2000)

- [6] Schmid, B.: Elektronische Märkte. In *Wirtschaftsinformatik*, Vol. 35, Nr. 5, 1993, 1993 465-480; Benjamin, Robert, Wigand, Rolf: *Electronic Markets and Virtual Value Chains on the Information Superhighway*. In: *Sloan Management Review*, Winter (1995) 62 - 72.; Choi, S.Y., D.O. Stahl, and A.B. Whinston: *The Economics of Electronic Commerce*. Macmillan Technical Publishing (1997); Shapiro, C., Varian, H.R.: *Information Rules. A Strategic Guide to the Network Economy*, Boston (1998)
- [7] Picot, A., Reichwald, R., Wigand, R.: *Die grenzenlose Unternehmung*, 4. Auflage, Wiesbaden (2000)
- [8] Forsa study viewed at <http://www.berlinonline.de/wissen/computer/internet/.html/200011/net01105.html>, 3-5-2001
- [9] In fact, a trend similar to the times of the LP, when printing costs for booklets increased in contrast to the production cost of the CD itself
- [10] Albers S., Clement M., Skiera B.: *Wie sollen die Produkte vertrieben werden? – Distributionspolitik*. In: Albers S., Clement M. et al *E-Commerce – Einstieg, Strategie und Umsetzung im Unternehmen*. Frankfurt (1999) 79-94
- [11] Zerdick, A. et al.: *Die Internet-Ökonomie – Strategien für die digitale Wirtschaft*. Springer, Berlin, Heidelberg (1999)
- [12] Heinrich, J.: *Medienökonomie.*, Vol. 1 Opladen Westdt. Verlag (1994)
- [13] Albers S., Clement M., Skiera B.: *Wie sollen die Produkte vertrieben werden? – Distributionspolitik*. In: Albers S., Clement M. et al *E-Commerce – Einstieg, Strategie und Umsetzung im Unternehmen*. Frankfurt (1999) 79-94
- [14] DeCSS
- [15] Gurley, W.: Digital music: The real law is Moore's law, *Fortune*, New York, Oct 2, 2000,, Volume: 142, Issue: 7 (2000) 268f.
- [16] Becker, A., Ziegler, M.: *Wanted: A survival plan for the music industry – Napster and the consequences*. Diebold Study, Munich (2000)
- [17] Seidel, N.: *Rundfunkökonomie: Organisation, Finanzierung und Management von Rundfunkunternehmen*. Wiesbaden (1993)
- [18] Tomczak, T., Schoegel, M., Birkhofer, B.: *Online-Distribution als innovativer Absatzkanal*. In: Bliemel, F. et al: *E-Commerce: Herausforderungen – Anwendungen – Perspektiven*. Wiesbaden, Gabler, 2000 219-238; Zerdick, A. et al.: *Die Internet-Ökonomie – Strategien für die digitale Wirtschaft*. Springer, Berlin, Heidelberg (1999)
- [19] Sinnreich, A.: *Digital Music Subscriptions: Post-Napster Product Formats*, Jupiter New York (2000)
- [20] VNU Entertainment Marketing Solutions: *Measuring the Influence of Music File Sharing*. New York (2000)
- [21] National Research Council, Committee on Intellectual Property Rights and the Emerging Information Infrastructure: *The Digital Dilemma – Intellectual Property in the Information Age*. National Academy Press, Washington, D.C. (2000)
- [22] Shapiro, C., Varian, H.R.: *Information Rules. A Strategic Guide to the Network Economy*, Boston (1998)
- [23] Zerdick, A. et al.: *Die Internet-Ökonomie – Strategien für die digitale Wirtschaft*. Springer, Berlin, Heidelberg (1999)
- [24] Picot, A., Reichwald, R., Wigand, R.: *Die grenzenlose Unternehmung*, 4. Auflage, Wiesbaden (2001)
- [25] Rifkin, J.: *The Future of Digital Music: Is There an Upside to Downloading?* Hearing Statements U.S. Senate Committee on the Judiciary viewed at http://www.senate.gov/~judiciary/7112000_jg.htm
- [26] Deutsch Bank UK: *New Media Mechanics - Value of Content Online*. London (2000)
- [27] National Research Council, Committee on Intellectual Property Rights and the Emerging Information Infrastructure: *The Digital Dilemma – Intellectual Property in the Information Age*. National Academy Press, Washington, D.C. (2000)

- [28] Black, L.: Understanding Consumer Demand to create business models that work, Webnoize Research, SGAE, Madrid, 25. 10. 2000
- [29] Heinrich, J.: Medienoekonomie., Vol. 2 Opladen Westdt. Verlag (1999)
- [30] Goldhammer, K., Zerdick, A.: Rundfunk Online – Entwicklung und Perspektiven des Internets fuer Hoerfunk- und Fernsehanbieter. Berlin (1999)
- [31] Piller, F. T.: Kundenindividuelle Massenproduktion. Die Wettbewerbsstrategie der Zukunft. München (1998)
- [32] National Research Council, Committee on Intellectual Property Rights and the Emerging Information Infrastructure: The Digital Dilemma – Intellectual Property in the Information Age. National Academy Press, Washington, D.C. (2000)
- [33] Hull, G.P., Greco, A.P., Martin, S.: The Structure of the Radio Industry, in: Greco, A.: The Media and Entertainment Industries. Readings in Mass Communications, Boston (2000) 122-156
- [34] According to Forrester Research, 90 percent of the music companies are working with DRM-providern, 60 percent in publishing and 50 percent in the film industry. Schreier, E.: Content out of Control, The Forrester Report, Cambridger, MA, September (2000)
- [35] Picot, A., Reichwald, R., Wigand, R.: Die grenzenlose Unternehmung, 4. Auflage, Wiesbaden (2000)
- [36] Goldhammer, K., Zerdick, A.: Rundfunk Online – Entwicklung und Perspektiven des Internets fuer Hoerfunk- und Fernsehanbieter. Berlin (1999)
- [37] Tang, Puay: How Electronic Publishers are Protecting against Privacy: Doubts about Technical Systems of Protection The Information Society v. 14, n. 1 Jan-Mar (1998) 19-31
- [38] Specifications for such an infrastructure is currently designed by the Secure Digital Music Initiative. www.sdmi.org SDMI, Document Nr. pdwg99070802, „SDMI Portable Device Specification Part 1, Version 1.0“, p. 21
- [39] Bakos Y., Brynjolffson E.: Bundling information Goods: Pricing, profits and Efficiency. Working Paper, Boston (1999)
- [40] Picot, A., Reichwald, R., Wigand, R.: Die grenzenlose Unternehmung, 4. Auflage, Wiesbaden (2000)
- [41] Zerdick, A. et al.: Die Internet-Ökonomie – Strategien für die digitale Wirtschaft. Springer, Berlin, Heidelberg (1999); Sinnreich, A.: Digital Music Subscriptions: Post-Napster Product Formats, Jupiter New York (2000)
- [42] Mori, R.: Superdistribution: The Conecpt and the Architecture. The Transactions of the IEICE E73, No 7. (1990) ; Cox, Brad: Superdistribution: Objects as Property on the Electronic Frontier. Addison-Wiley (1996); Morin, Jean-Henry (1999) p. 22. It seems as if in parallel Brad Cox has developped a similar system, that is documentet in 1994 with his system, CopyFree Software
- [43] Morin, Jean-Henry: Commercial Electronic Publishing over Open Networks: A Global Approach Based on Mobile objects (Agents). Dissertation Universitaet Genf (1999)
- [44] Tang, Puay: How Electronic Publishers are Protecting against Privacy: Doubts about Technical Systems of Protection The Information Society v. 14, n. 1 Jan-Mar (1998) 19-31
- [45] Albers S., Clement M., Peters K.: Marketing mit Interaktiven Medien. Strategien zum Markterfolg, Frankfurt am Main (1998)
- [46] The classiv example are stock quotes.
- [47] Thurow, N.: Die digitale Verwertung von Musik aus der Sicht von Schallplattenproduzenten und ausübenden Künstlern, in: Becker, Jürgen / Dreier, Thomas, Urheberrecht und digitale Technologien, Vortragssammlung der Sitzung des Instituts für Urheber- und Medienrecht, UFITA-Schriftenreihe, Baden-Baden (1994) 77; Zerdick, A. et al.: Die Internet-Ökonomie – Strategien für die digitale Wirtschaft. Springer, Berlin, Heidelberg (1999)

Author Index

Aberer, K.	142	Laender, A.H.F.	154
Abrahão, S.	16	Lam, T.-W.	216
Abrazhevich, D.	81	Lang, F.	206
Agrawal, D.	250	Lee, B.-R.	176
Amamiya, M.	305	Lee, H.S.	240
Amamiya, S.	305	Lee, J.	240
Arantes, A.R.	154	Lenkov, D.	270
Banaei-Kashani, F.	280	Li, W.-S.	250
Bodendorf, F.	206	List, B.	315
Brisaboa, N.R.	29	Loney, F.N.	295
Buhse, W.	337	Magkos, E.	186
Candan, K.S.	250	Neuhold, E.J.	326
Cannataro, M.	132	Nicolle, C.	260
Chang, K.-A.	176	Omelayenko, B.	226
Cheung, D.	216	Park, S.	49
Chitchyan, R.	39	Pastor, O.	16
Chrissikopoulos, V.	186	Penabad, M.R.	29
Christoffel, M.	101	Places, Á.S.	29
Chung, T.-S.	49	Po, O.	250
Cuzzocrea, A.	132	Pugliese, A.	132
Dani, A.R.	91	Pulvermueller, E.	39
Das, A.	121	Qiu, Z.	326
Deschner, D.	206	Quirchmayr, G.	315
Dorneles, C.F.	60	Radha Krishna, P.	91
Eder, J.	71	Rashid, A.	39
Faisal, A.	280	Rebstock, M.	196
Faruque, J.	280	Rodríguez, F.J.	29
Fensel, D.	226	Sahai, A.	270
Fons, J.	16	Shahabi, C.	280
Franke, T.	101	da Silva, A.S.	154
Golgher P.B.	154	Speck, A.	39
Graupner, S.	270	Strametz, W.	71
Han, S.-Y.	49	Sung, W.-K.	216
Helmy, T.	305	Tjoa, A.M.	315
Hemmje, M.	326	Tomsich, P.	166
Heuser, C.A.	60	Vittori, C.M.	60
Ho, W.-S.	216	Wang, G.	121
Hsiung, W.-P.	250	Wang, P.	240
Katzenbeisser, S.	166	Werthner, H.	1
Kim, H.-J.	49	Wombacher, A.	142
Kim, T.-Y.	176	Yang, D.	216
Kim, W.	270	Yé tongnon, K.	260
Kotkamp, S.	101	Yiu, S.-M.	216
Kotzanikolaou, P.	186	Yu, C.-C.	111